

Tudományos Diákköri Konferencia

Kondor Péter

V. éves építészhallgató

**Fedélidomok és ácsszerkezeti részletek szerkesztése
dinamikus geometriai programmal**

Konzulens:

Dr. Szoboszlai Mihály

egyetemi docens

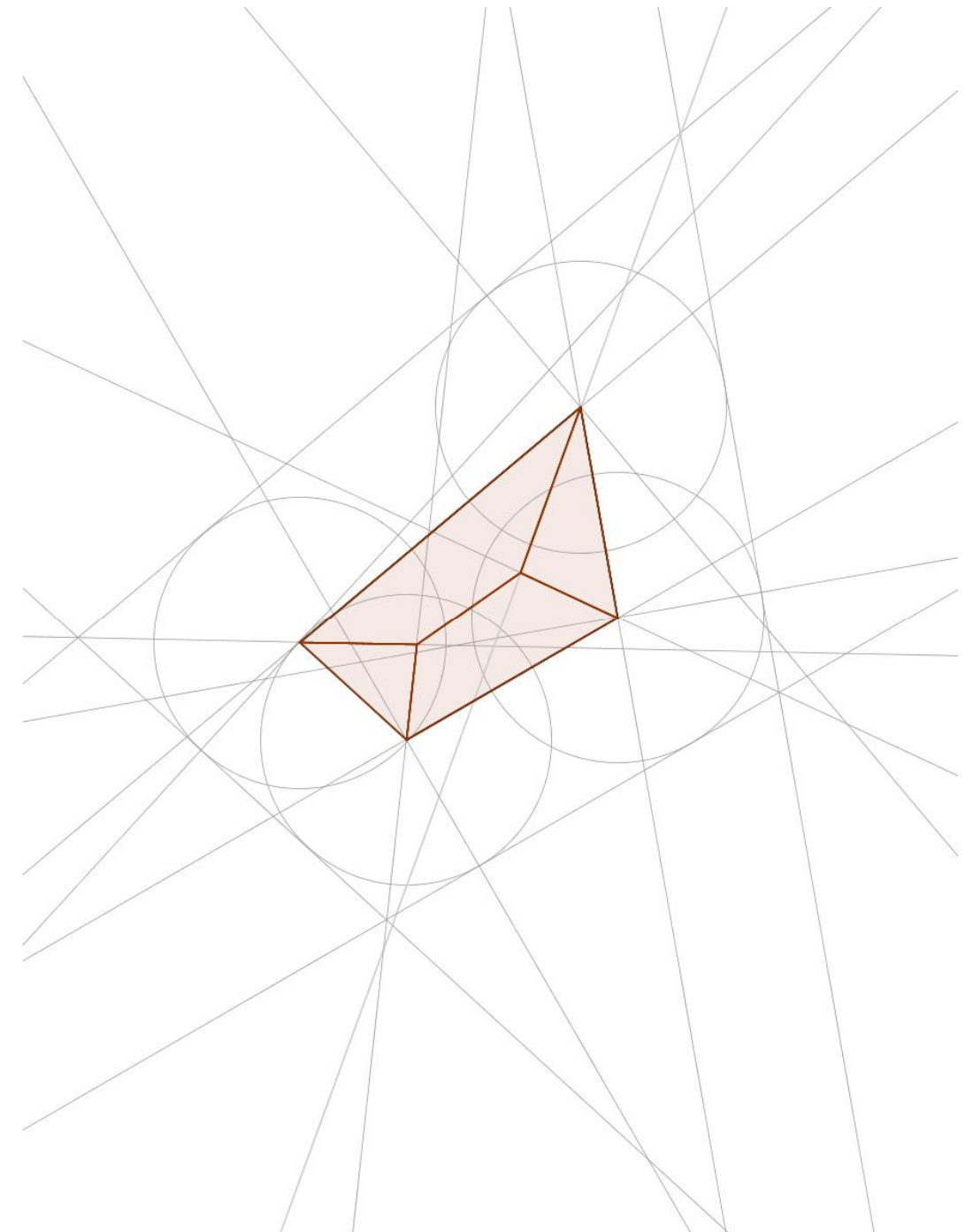
Budapesti Műszaki és Gazdaságtudományi Egyetem

Építészeti Ábrázolás Tanszék

2011

Tartalomjegyzék

1. Bevezetés	3
1.1 A téma evolúciója	3
1.2 A fedélszerkesztési eljárásokról általában.....	4
2. Dinamikus geometriai programok	5
2.1 A Geogebra, mint dinamikus geometriai program.....	5
2.2 A program felhasználói felülete.....	5
3. Szerkesztéseimről általában	6
4. Részletrajzok.....	7
4.1 Gúla – csúcs szerkesztése	7
4.2 Döntött szelemenek kapcsolata.....	8
4.3 Csonkaszaruk szerkesztésének egy esete	9
5. Fedélidom-szerkesztések.....	12
5.1 Tetőközepelések és az alkalmazott algoritmus bemutatása.....	12
5.2 Komplex fedélidom-szerkesztés	15
6. Összegzés.....	16
6.1 Elért eredmények	16
6.2 Rendszerkövetelmények, felhasználhatóság.....	16
6.3 További terveim	16
7. Irodalomjegyzék	17
8. Egy szerkesztés forráskódja	18



1. Bevezetés

1.1. A téma evolúciója

Jelen dolgozatom célja többek között, hogy munkámmal valamilyen új, használható és hasznos dolgot hozzak létre. Így tértem vissza a fedélszerkezetekkel kapcsolatos korábbi tanulmányaimhoz és tapasztalataimhoz, melyekről a következőkben számolok be.

Első benyomásaim, melyek választott témámat adták, az Épületszerkezettan 2 tantárgy keretein belül értek. Ekkor foglalkoztunk többek között a fedélidomok szerkesztésével és a tetőszerkezetek konstruálásával. Elsajátítottuk a fedélidomok szerkesztésének különféle szabályait, megtanultuk a fedélszerkezetek számos típusának felépítését és működését, valamint megismerkedtünk az ezen szerkezetekben alkalmazott ács- és mérnök jellegű kötésekkel. Ugyanakkor nem tanultunk olyan szabályt, amely alapján bármely fedélidom egységesen végigszerkeszthető lenne, és nem foglalkoztunk mélyebben az él- és vápaszaruk geometriájával. Ezen témakörök iránt már ekkor elkezdtem komolyabban érdeklődni.

Második benyomásom a Szerves építészet 2 tantárgy keretein belül szervezett sárospataki építőtáborban ért. A tábor során egy gimnázium udvarát hoztuk helyre. Ennek részeként felépítettünk egy szabályos ötszög alaprajzú pavilont (1. és 2. ábra). Ennek az épületnek számos csomópontját térben, egymáshoz rendhagyó szögekben csatlakozó fa szerkezeti elemek képezték. A valódi problémát az jelentette, hogy az elemek lesabálásához szükséges sablonok egy véletlen során nem jutottak el időben az építkezés helyszínére. Így ezeket a helyszínen kellett kiszerkeszteni és legyártani. Mindez a megfelelő eszközök hiányában igen körülményes munkát jelentett. Szerkesztéseinket nem tudtuk kellő pontossággal elvégezni és voltak olyan esetek is, amikor csak közelítő számításokra tudtunk hagyatkozni. Ez pedig utólagos pontosításokat és számos plusz munkaórát vont maga után. Jelen dolgozatom témája ezen tapasztalataimból született meg.

Kezdetben olyan eljárásokat, rövid szerkesztéseket hoztam létre, amelyekkel a térben, rendhagyó szögben találkozó elemek szabásmintája akár a helyszínen is viszonylag könnyedén kiszerkeszthető. Ezeket az eljárásokat végül egy segédlet formájában gondoltam közzétenni. Ennek a segédletnek készítettem el egy nem végleges változatát, amit bemutattam a konzulensemnek.

Munkám lényegi része ekkor következett. Konzulensem javaslatára újabb, komplex geometriájú – elsősorban mérnök jellegű - fa kapcsolatokat kerestem különféle könyvekből, segédletekből. Az így talált szerkezeti részleteket egy dinamikus geometriai programban szerkesztettem ki, az elemekről pedig kiforgatott, méretekkel ellátott rajzokat készítettem. Ezeket a szerkesztéseket úgy végeztem el, hogy azok – a kiindulási méretek és szögek állításával - konkrét esetekre adaptálhatók legyenek. Ezzel párhuzamosan hasonló tevékenységet végeztem a fedélidom-szerkesztés témakörében is, melyet a továbbiakban részletezek. Végül az elvégzett szerkesztéseket egy honlap keretében tettem elérhetővé.



1. ábra Sárospatak, építőtábor, 2010. augusztus – a pavilon egy korai építési stádiumban

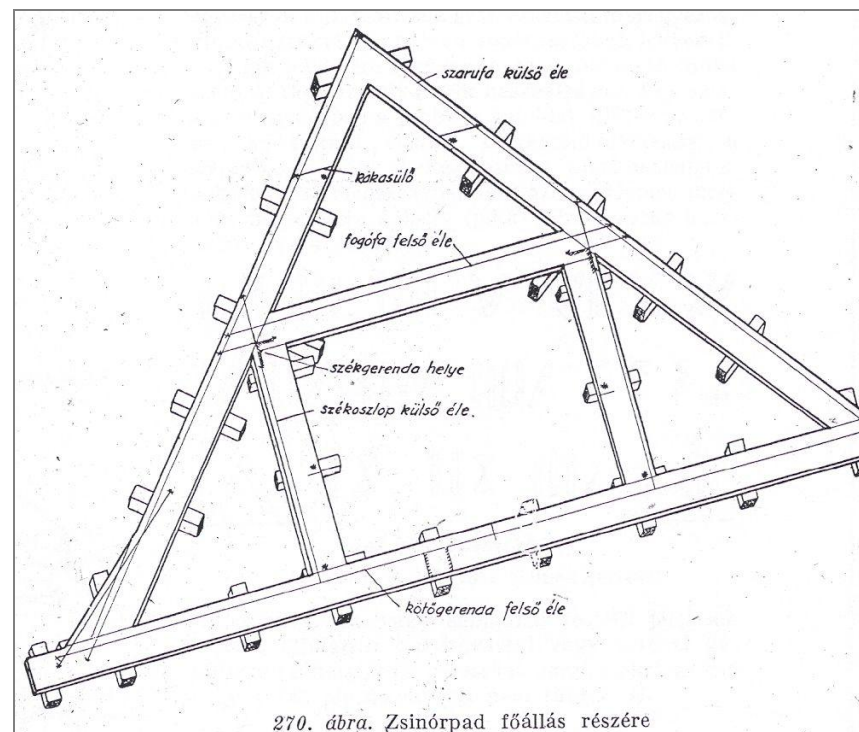


2. ábra Sárospatak, építőtábor, 2010. augusztus – a pavilon egy késői építési stádiumban

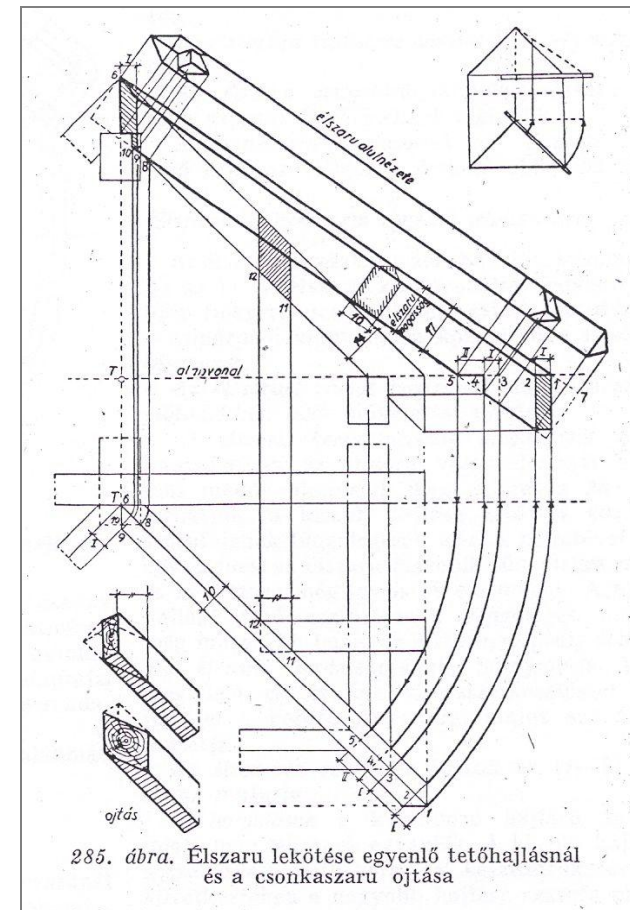
1.2. A fedélszerkesztési eljárásokról általában

Az ácsok hagyományosan zsinórpadon végzik el a fedélszerkezetek elemeinek kiserkesztését és a fedélszék előzetes összeszerelését. (3. 4. 5. 6. ábra, [4], 17. old.) Ma már nem mindenhol alkalmaznak zsinórpadot. Ennek oka egyrészt, hogy az ács jellegű kötéseket a mérnök jellegű kötések váltották fel, melyek geometriája az előzőeknél egyszerűbb. Másik oka, hogy az összetettebb fedélszékek elemeit számítógéppel rendszerint előre megtervezik és azokat az így készült terv vagy sablon alapján legyártják. Akadnak olyan esetek is, amikor egyszerűen a pontosság rovására elhagyják a zsinórpadon való előzetes kiserkesztést.

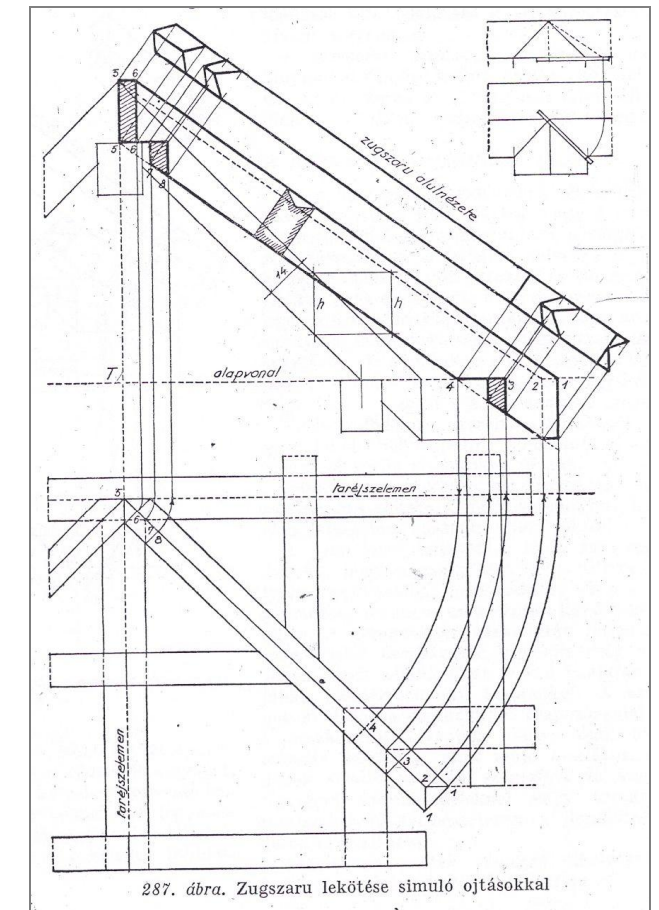
A zsinórpad lehetővé teszi olyan összetettebb geometriájú elemek, mint az él- és vápaszaruk kiserkesztését is. Ebben az ácsok általában nagy jártassággal rendelkeznek, de sokszor szerkesztés nélkül, egyszerűen szemre vágják le ezeket az elemeket. Ebből kifolyólag ezekkel a viszonylag könnyen kezelhető esetekkel dolgozatomban nem foglalkozok.



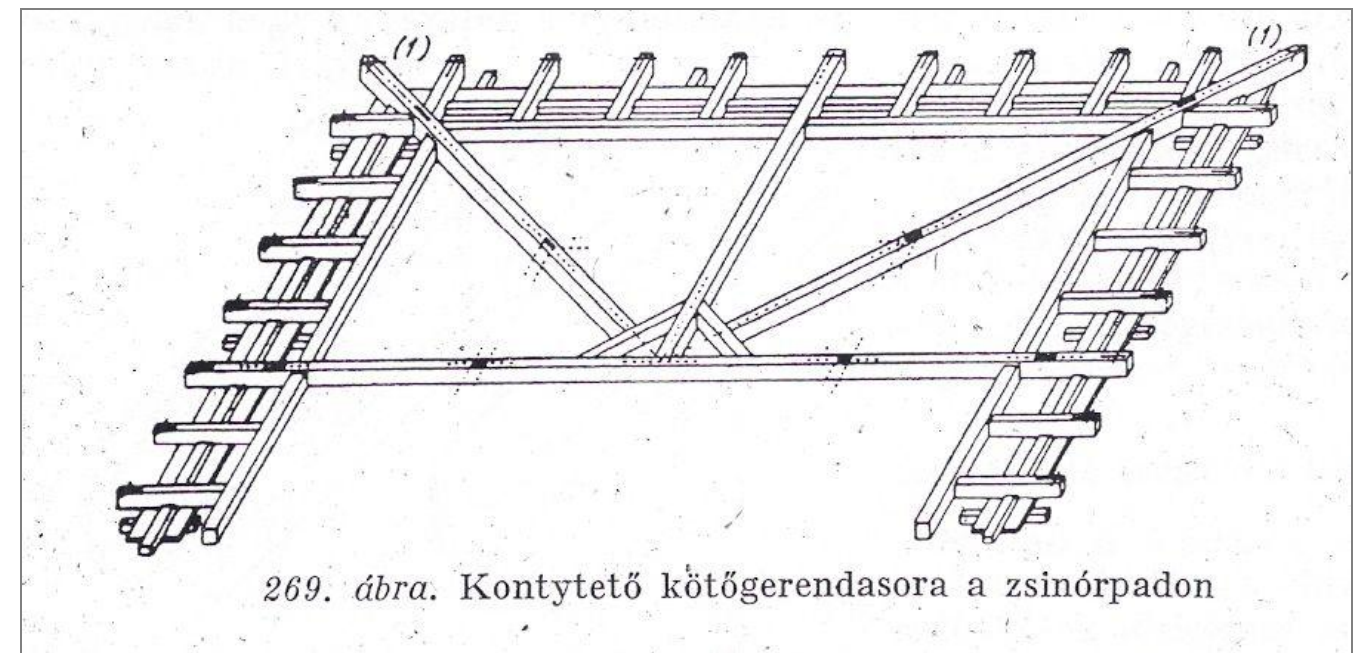
3. ábra Zsinórpád főállás részére



4. ábra Élszaru lekötése zsinórpadon



5. ábra Vápaszaru (régén zugszaru) lekötése zsinórpadon



6. ábra Kontytető kötőgerenda-sora a zsinórpadon (régi szerkezeteknél)

2. Dinamikus geometriai programok

2.1. A Geogebra, mint dinamikus geometriai program

A dinamikus, vagy más szóval interaktív geometriai programok lényege, hogy a velük végrehajtott szerkesztés nem a pillanatnyi statikus ábrával azonos, hanem a szerkesztési struktúrát, az objektumok leszármazási rendszerét jelenti. Az alakzatok egy része szabadon mozgatható, de vele együtt az összes olyan alakzat helye is változni fog, amelyeket belőle kiindulva szerkesztettünk. A legismertebb ilyen programok: Cabri, Cinderella, Euklid, Euklides, GeoGebra, Geometer's Sketchpad ([6], 17. old.).

Választásom a következők miatt esett a Geogebra:

- Ebben érhető el a legtöbb funkció
- Ingyenes
- Jelenleg is fejlesztés alatt áll – Legújabb, kész verziója a Geogebra 4.0, de letölthető már a Geogebra 5.0 béta verzió is, mely 3D-s szerkesztési lehetőséget is tartalmaz.
- A vele elkészített szerkesztések weblapként menthetők. Megnyitásukhoz szükséges: egy webböngésző és megfelelő verziójú, telepített Java szoftver.

A program fejlesztését egy nemzetközi csapat végzi, többek között magyar szakemberek bevonásával is. A csapat vezetője az Ausztriában élő Markus Hohenwarter ([5], 17. old.).

2.2. A program felhasználói felülete

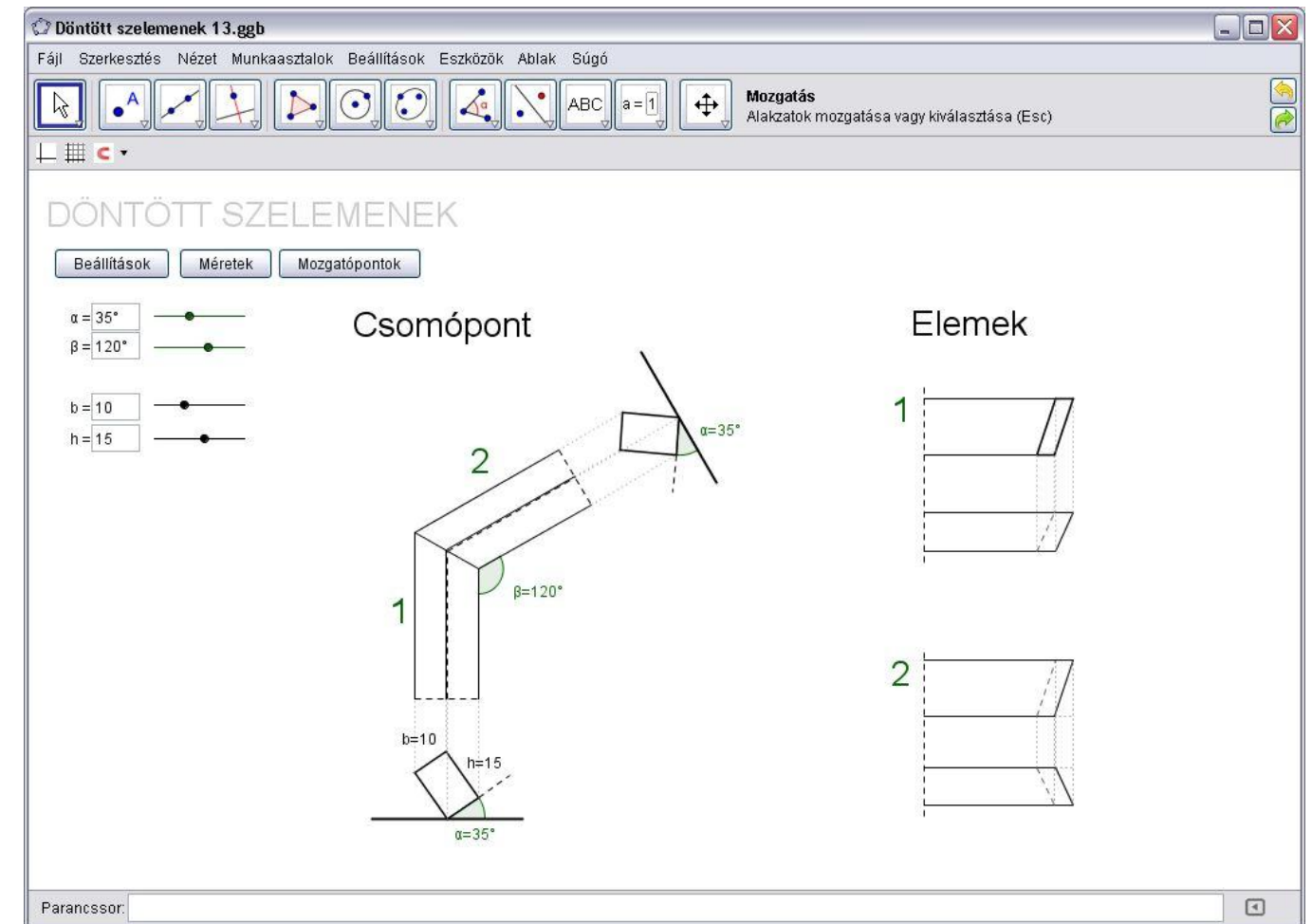
A Geogebra egyszerű, könnyen átlátható felhasználói felülettel rendelkezik. A fent megjelenő ikonok mindegyike egy-egy legördülő menüt rejt, melyekben hasonló rajzi elemek, funkciók érhetők el (8. ábra). A képernyő alapvetően két részre oszlik: egy algebra- és egy geometria ablakra. A geometriaablakban rajzolt elemek mindegyike az algebra ablakban is megjelenik matematikai képletével leírva.

Az általam a szerkesztések során leggyakrabban használt rajzi elemek:

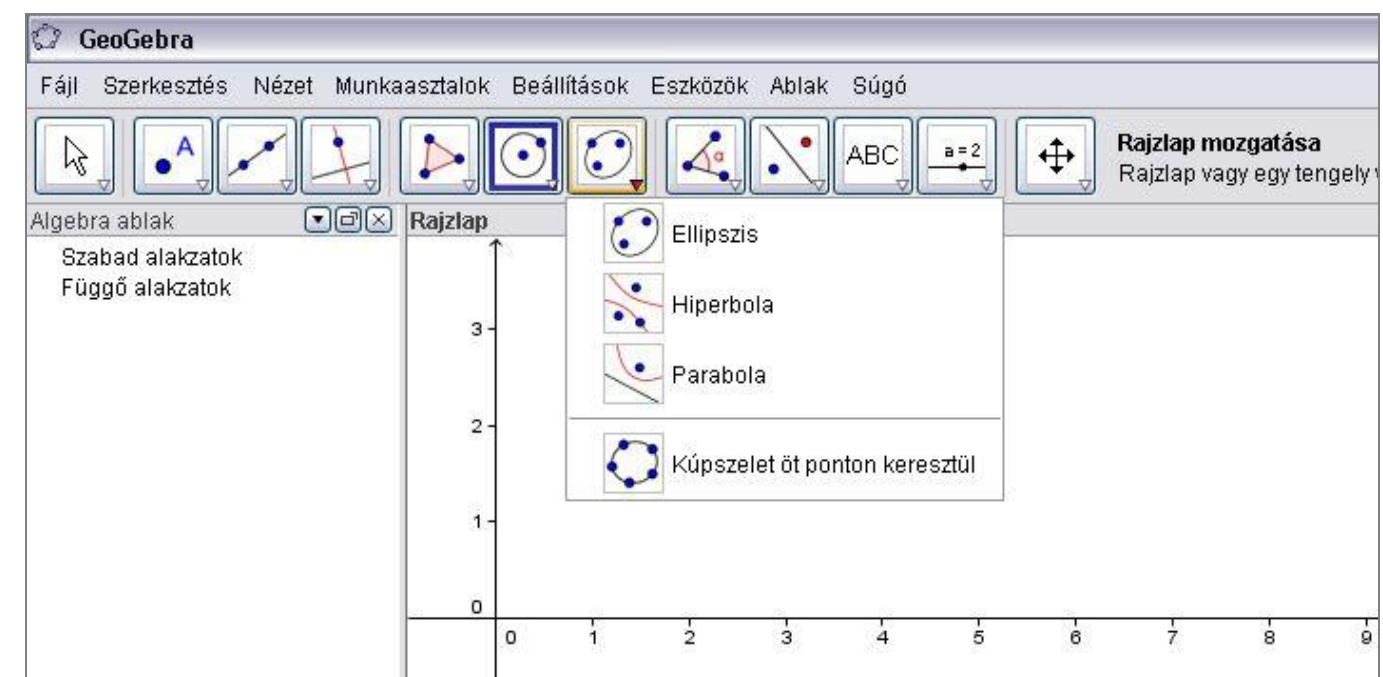
- pontok rajzolása
- egyenesek rajzolása adott pontokon keresztül / más egyenesekkel párhuzamosan
- körök rajzolása adott középponttal és sugárral / adott középponttal és ponttal
- szögek rajzolása 3 ponttal / adott mérettel

A leggyakrabban használt egyéb eszközök:

- szöveg beszúrása
- csúszkák egy változó értékének a megadásához (ezen keresztül a kész rajz szemléletes módosításához)
- jelölőnégyzetek és gombok az alakzatok, funkciók elrejtéséhez és megjelenítéséhez
- beviteli mezők valamely változó értékének a megadásához – ezeket rendszerint a csúszkákkal együtt használtam



7. ábra Egy kész szerkesztés állítható szögekkel és méretekkel – Geogebra 4.0



8. ábra Eszközök használata a Geogebra-ban

3. Szerkesztéseimről általában

Mint korábban utaltam rá, a dinamikus geometriai programokkal végrehajtott szerkesztésekben az alakzatok egy része szabadon, akár utólag is mozgatható. Így olyan szerkesztéseket készíthettem, melyeket később bárki a saját esetére képes adaptálni. Ezeket a szerkesztéseket a következő két fejezetben (Részletrajzok és Fedélidom-szerkesztés címen) mutatom be részletesen.

Munkámat két fő irányvonalon végeztem:

- Egyrészt olyan mérnök jellegű fa kapcsolatokat vettem számba és szerkesztettem ki a Geogebra segítségével, melyek a szokásosnál mélyebb geometriai tudást igényelnek – lásd: 3.2 Részletrajzok c. fejezet
- Másrészt pedig a fedélidomok szerkesztésére kerestem alternatív megoldást a program felhasználásával – lásd: 3.3 Fedélidom-szerkesztés c. fejezet

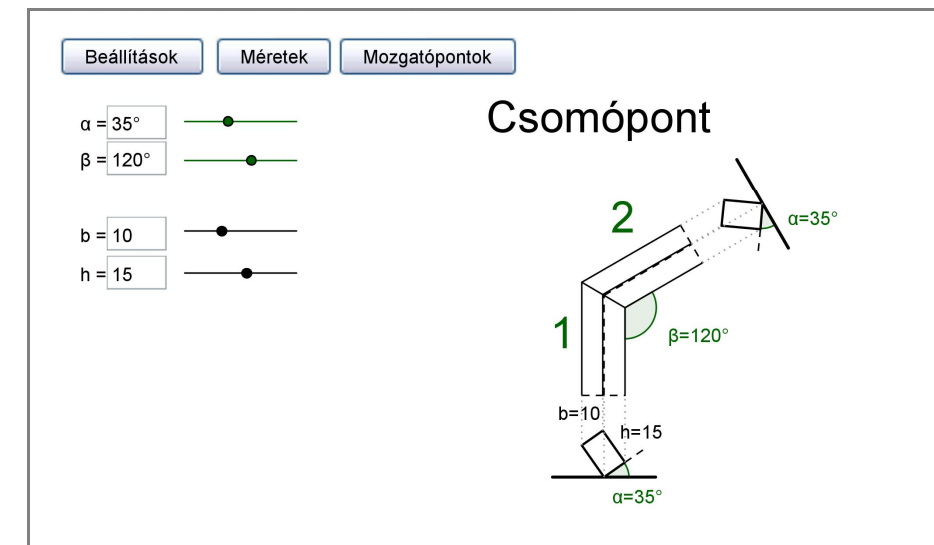
Szerkesztéseim egy részébe egy rövidebb (gombokból létrehozott) kezelőfelületet is beépítettem, mely lehetővé teszi egyes funkciók megjelenítését és elrejtését.

A beépített kezelőfelületeken általánosan alkalmazott funkciók:

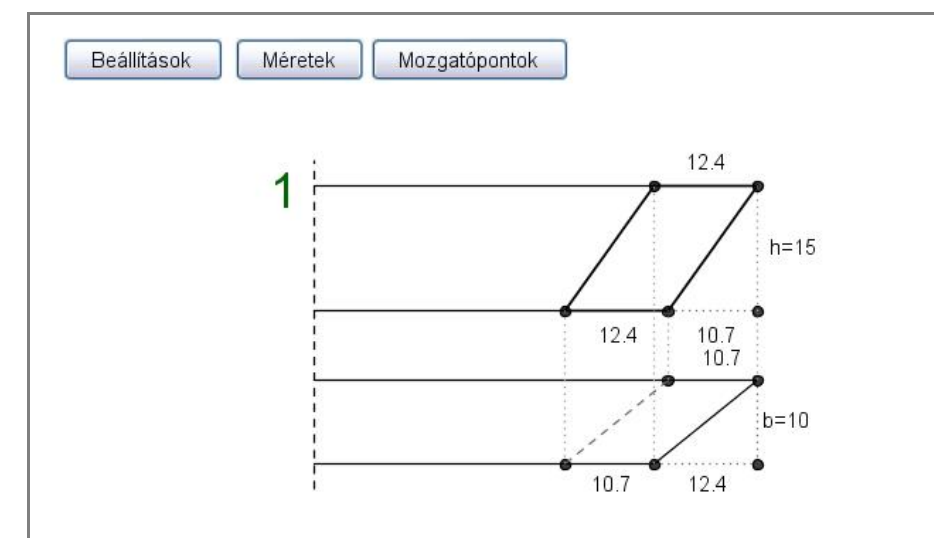
-*Beállítások (9. ábra):* Jellemzően az alakzatok méreteinek és szögeinek beállítási lehetőségeit tartalmazza csúszkákkel és beviteli mezőkkel.

-*Méretek vagy kótapontok (10. ábra):* Az egyes elemek, fedélidomok méreteit, területeit jeleníti meg vagy rejt el. A programban nincs beépített kótázási lehetőség, ezért a mért pontokat egy más, az ábrán bemutatott módon jelöltem.

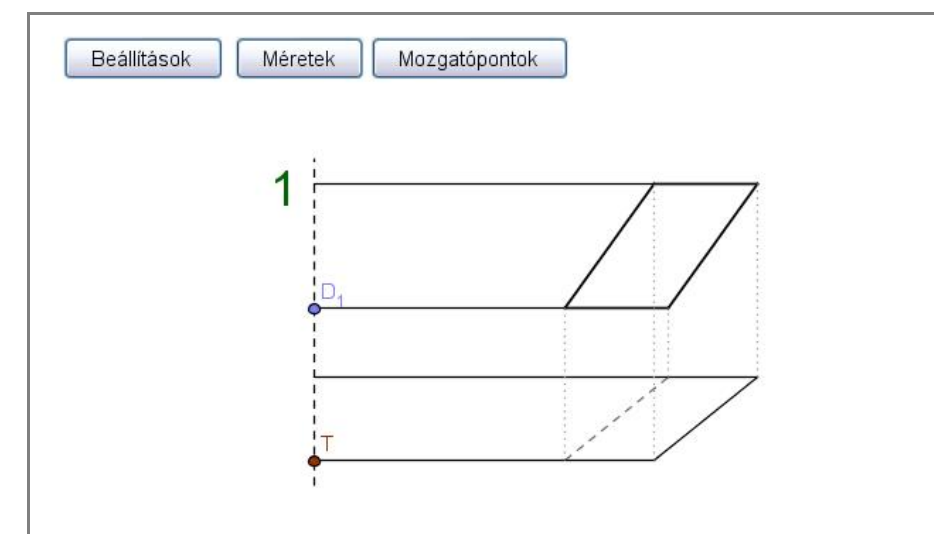
-*Mozgatópontok (11. ábra):* A mozgatópontok megjelenítéséért és elrejtéséért felel. A piros mozgatópontok egyszerre több elemet mozgatnak, a kék mozgatópontok pedig egy elemet, vagy egy elem valamely részletét.



9. ábra Beállítások megjelenítése



10. ábra Méretek megjelenítése



11. ábra Mozgatópontok megjelenítése

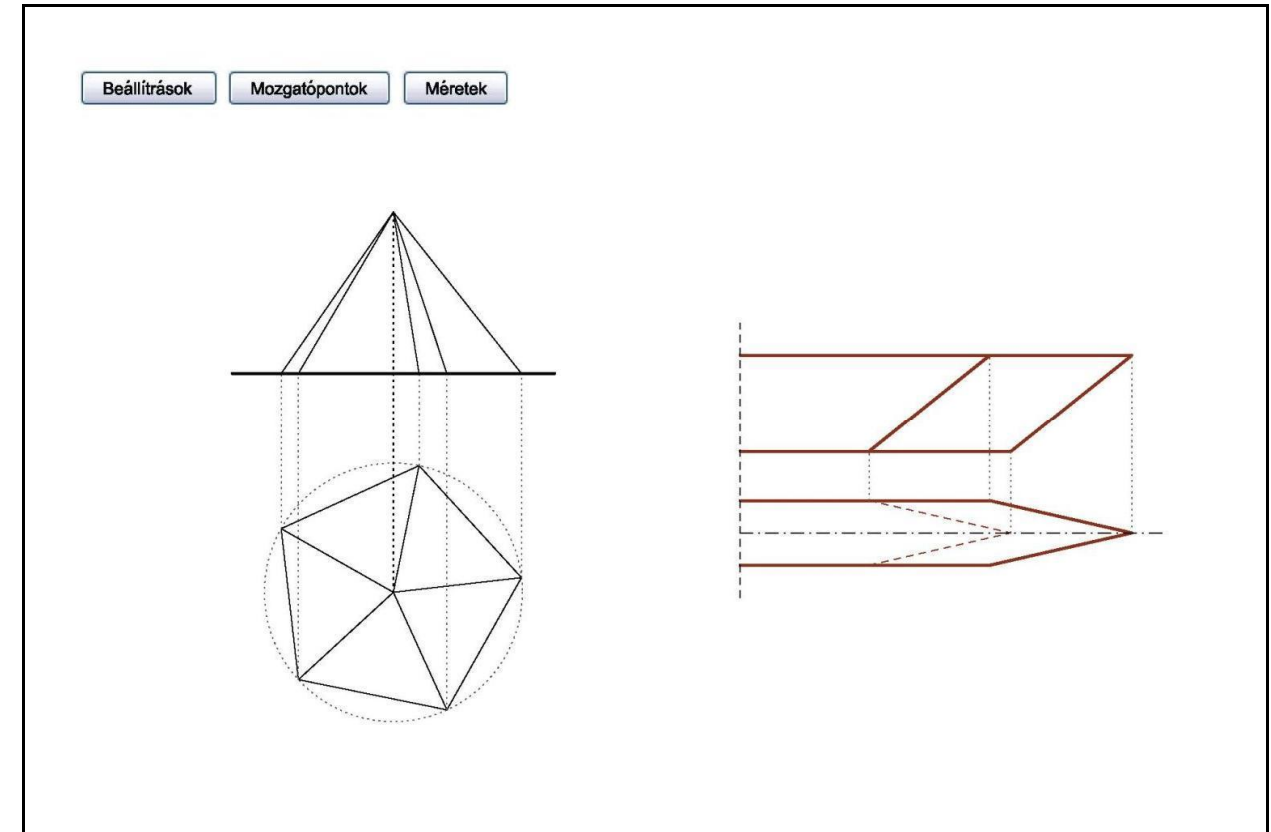
4. Részletrajzok

4.1. Gúla – csúcs szerkesztése

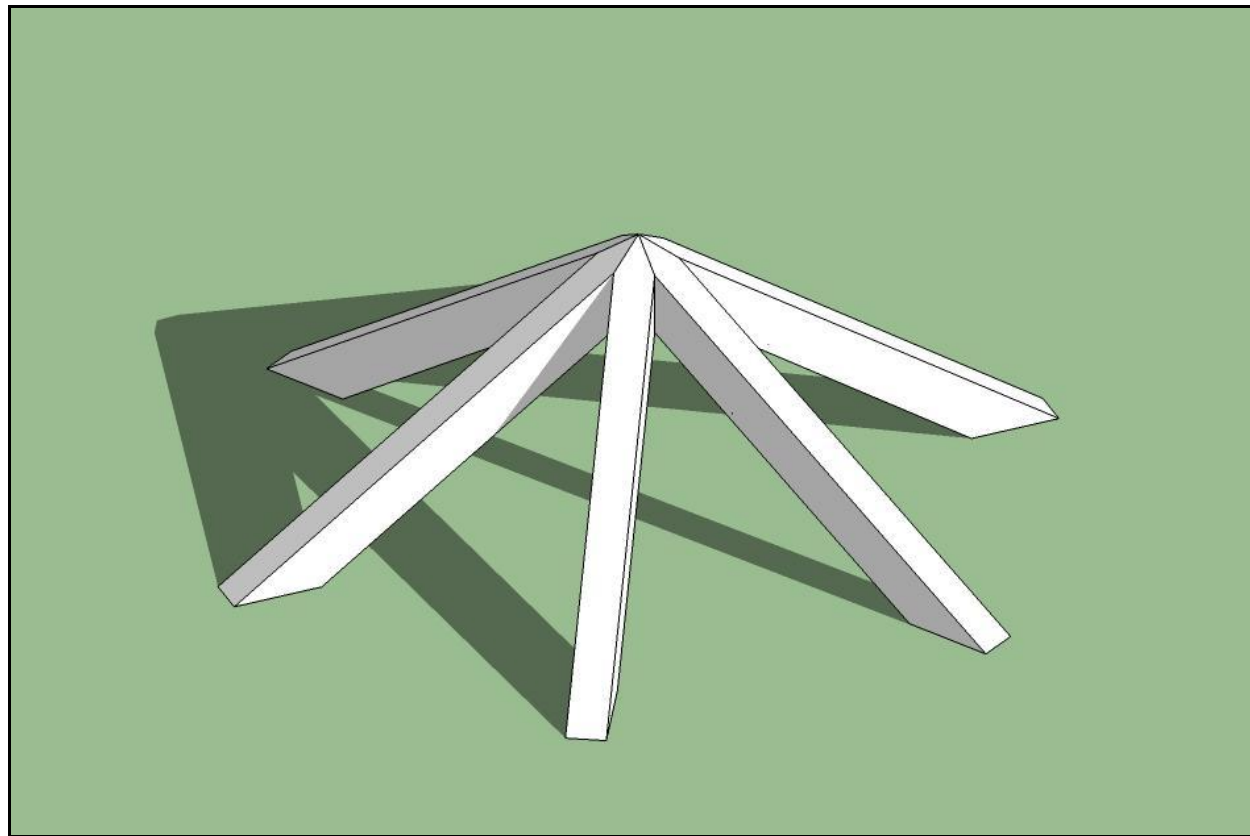
Munkám ezen részében szabályos gúla alakú tetőszerkezetek (tornyok, sáttortetők) élszaruvégződéseinek kiszerkesztését végeztem el a már említett dinamikus geometriai programban. (12. és 13. ábra). A szerkesztés elvégzéséhez a csonkaszaruk végéből vízszintes síkkal kimetszett éktestek élei, oldalai és szögei, valamint a gúla méretei közötti összefüggéseket használtam fel (14. ábra).

A szerkesztés lépései vázlatosan:

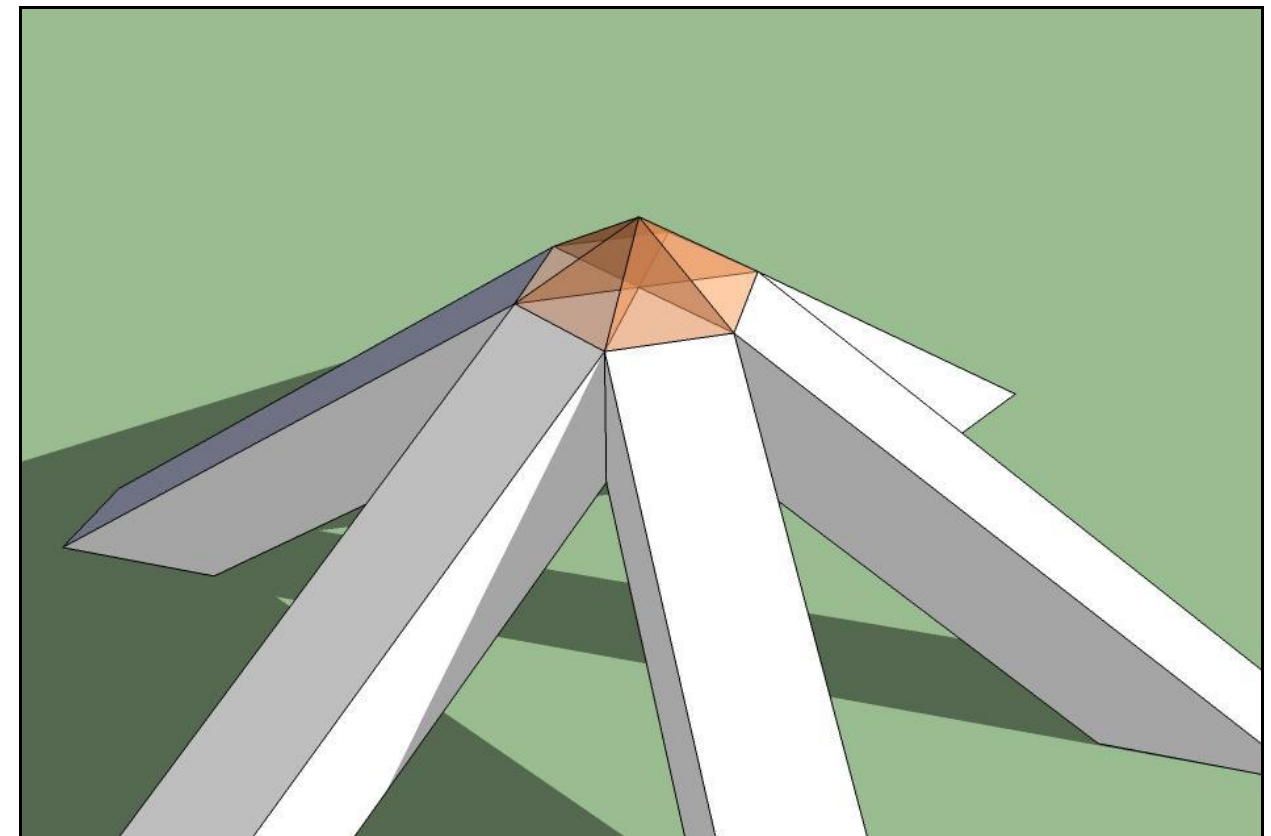
1. Beviteli mezőket és csúszkákat adtam meg, melyekkel a leendő felhasználó maga határozhatja meg a szerkesztés paramétereit, mint a gúla oldalszámát, magasságát, az alap köré írt kör sugarát valamint az élszaruk szélességi és magassági méreteit. Ezeket a beállítások fül alatt tettem elérhetővé.
2. Egy körbe különböző oldalszámú szabályos sokszögeket írtam, melyekre gúlákat szerkesztettem.
3. A beállított paraméterekhez kötöttem, hogy milyen oldalszámú gúla jelenjen meg, mekkora legyen a megjelenített gúla magassága és az alaprajz köré írható kör sugara.
4. Végül a beállított paraméterekhez kötve kiszerkesztettem az élszaruk végződéseit.



13. ábra Szabályos gúla alakú tetőszerkezet – az élszaruk végződéseinek kiszerkesztése



12. ábra Szabályos gúla alakú tetőszerkezet – élszaruk találkozása 3D-ben



14. ábra A szerkesztés alapját adó, kimetszett éktestek

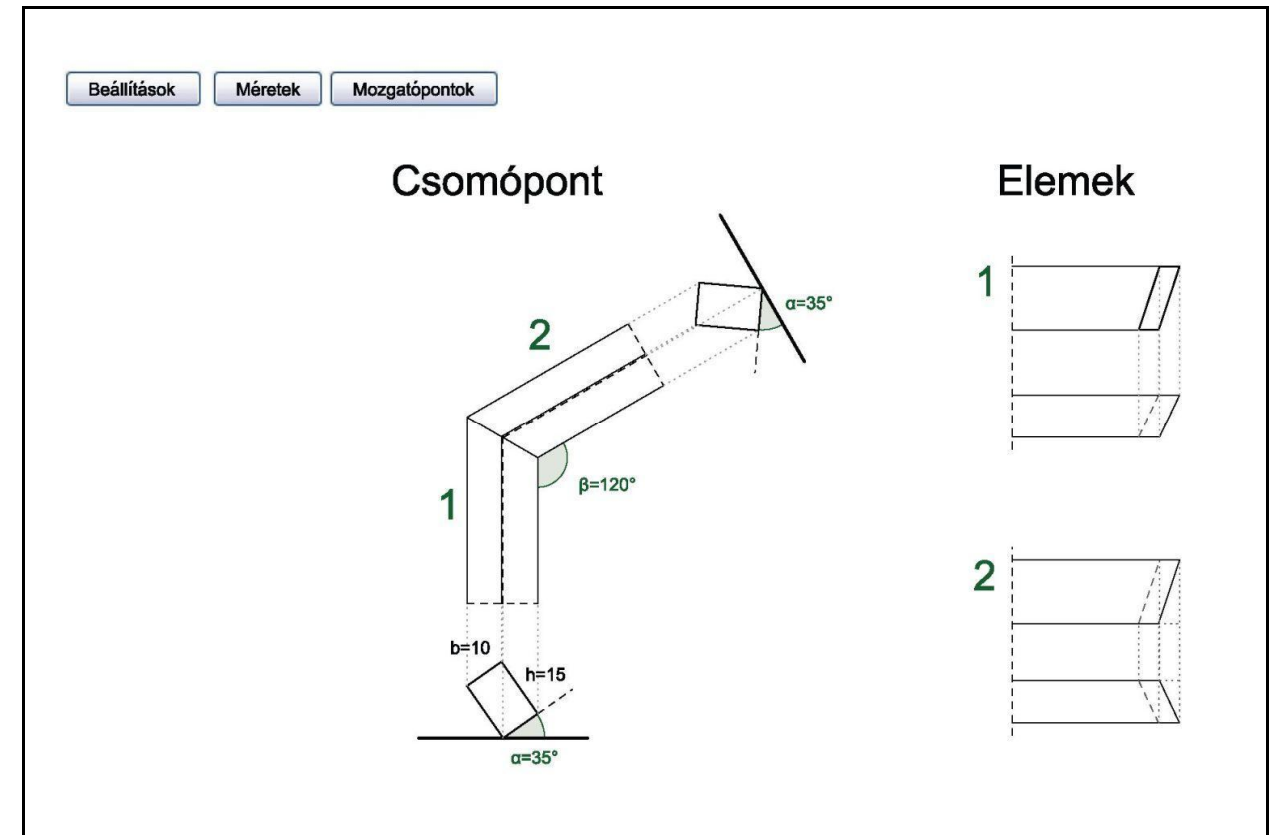
4.2. Döntött szelemenek kapcsolata

Ilyen típusú fa kapcsolattal a bevezetőben is említett építőtáborban találkoztam először. Pontosan értelmezve ott ezek az elemek nem szelemenként funkcionáltak, - mint jelen fejezet címében - hanem az egymáshoz illesztett keretek alsó és felső vízszintes éleit adták. A keretek összeillesztéséből született meg az épület vázszerkezete. (1. ábra)

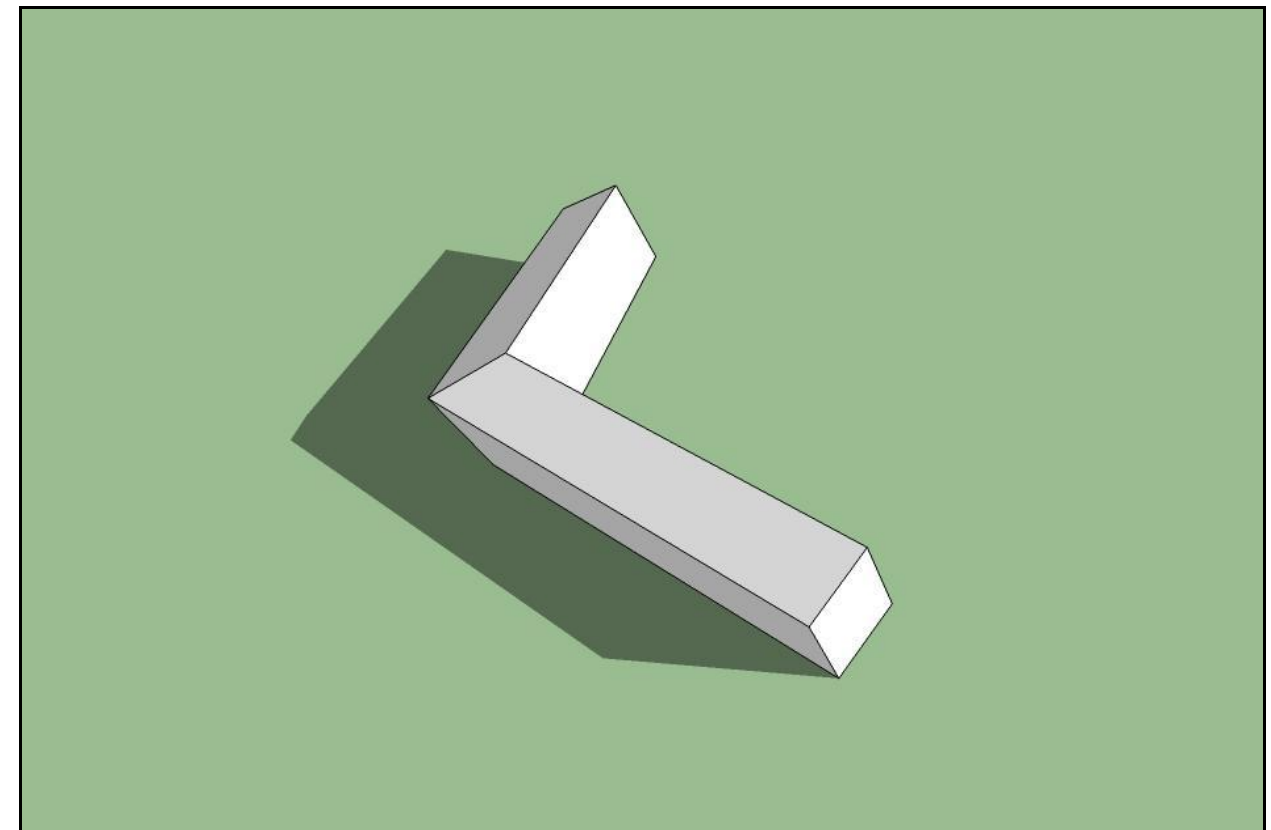
Ez volt az egyik olyan mérnök jellegű kapcsolat, amelyhez a helyszínen ki kellett szerkesztenünk és le kellett gyártanunk a sablonokat, amelyekkel a csatlakozó elemek végződése megfelelő geometriájúra szabható (15. és 16. ábra).

A programmal elkészített szerkesztésben beállítható méretek, szögek:

- A csatlakozó szelemenek szélessége és magassága
- A szelemenek alsó síkjának vízszintessel bezárt szöge
- A szelemenek alaprajzon egymással bezárt szöge



15. ábra Egymáshoz csatlakozó, azonos szögben döntött szelemenek találkozása és az egyes elemek végződése kiszerkesztve



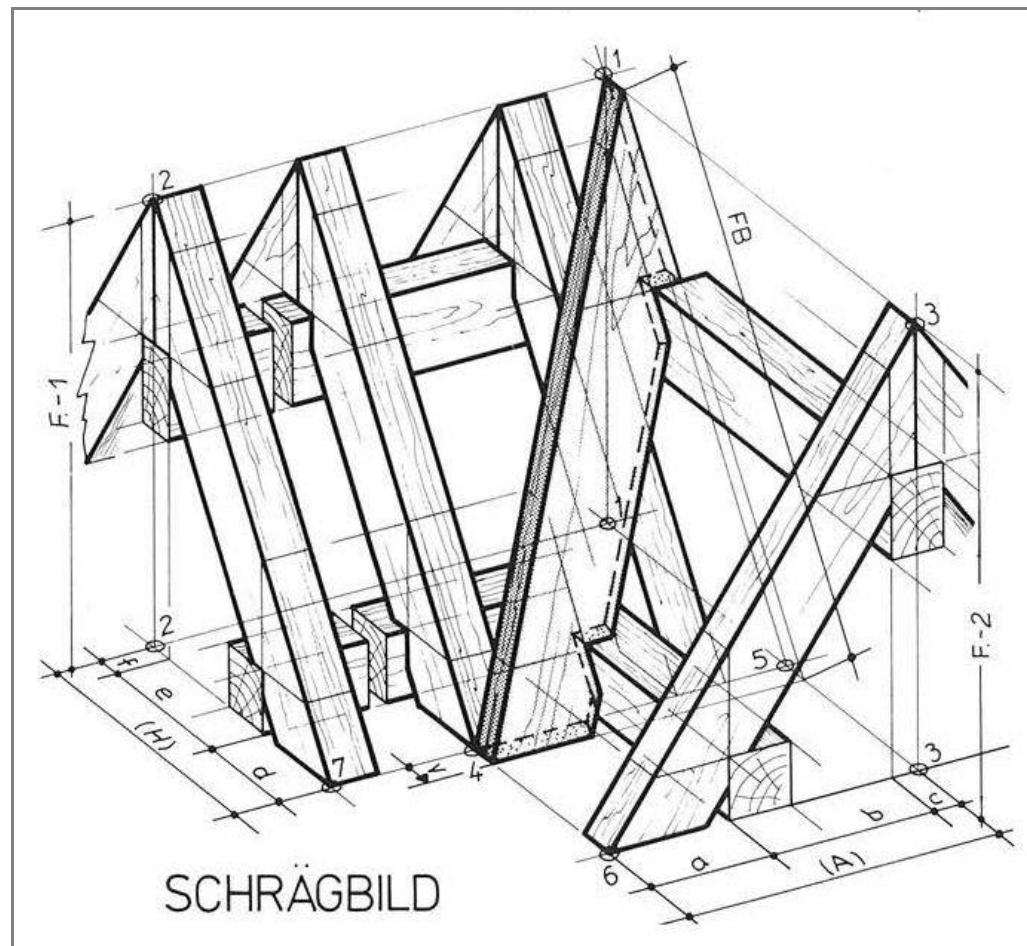
16. ábra Egymáshoz csatlakozó, azonos szögben döntött szelemenek találkozása 3D-ben

4.3. Csonkaszaruk szerkesztésének egy esete

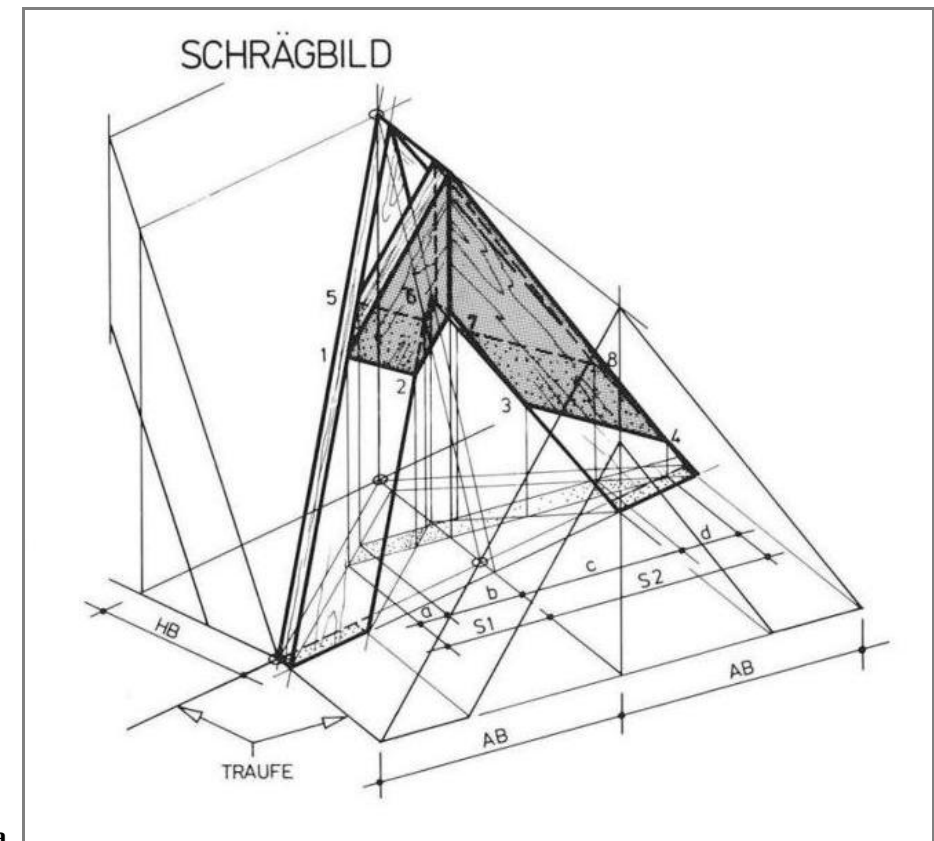
Jelen fejezet a csonkaszaruk szerkesztésének azon esetét tárgyalja, amikor egy végigmenő nyeregtetőbe egy másik, rendszerint kisebb gerincmagasságú nyeregtető csatlakozik. A becsatlakozó nyeregtető csonkaszarui a végigmenő tető szaruira fektetett pallóról vagy egy vápaszaru felső síkjáról indulnak. (17. 18. 19. ábra, [1], 17. old.)

Mivel ebben a szerkesztésben nem a taréj- vagy talpszelemennel való csatlakozás kiszerkesztése volt a célom, ezért ezeket az elemeket és a velük létrejövő kapcsolatot időmegtakarítás, a szerkesztés egyszerűsítése és a könnyebb átláthatóság végett nem ábrázoltam.

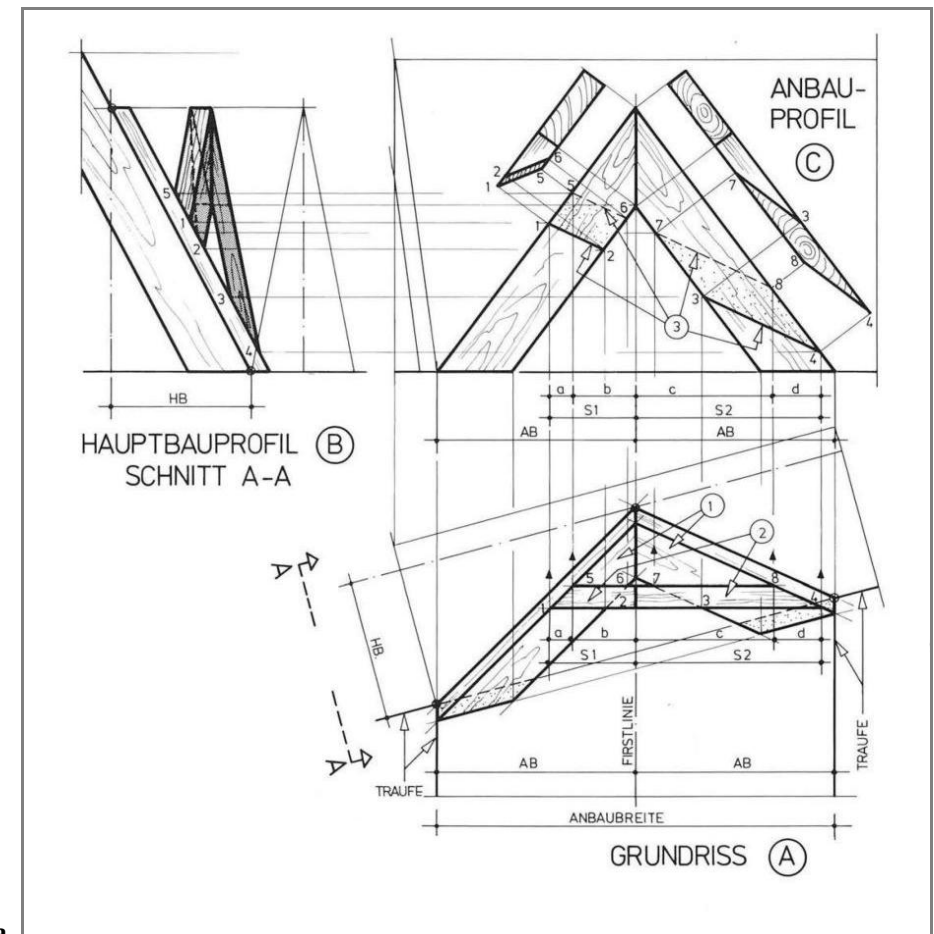
Erre a szerkesztésre előzőleg két kísérleti modellt is készítettem. Ezeket a bennük jelentkező, alapvető hiányosságok miatt félkész állapotban hagytam. A következőkben mind a kísérleti modelleket, mind a kész szerkesztést bemutatom.



17. ábra



18. ábra



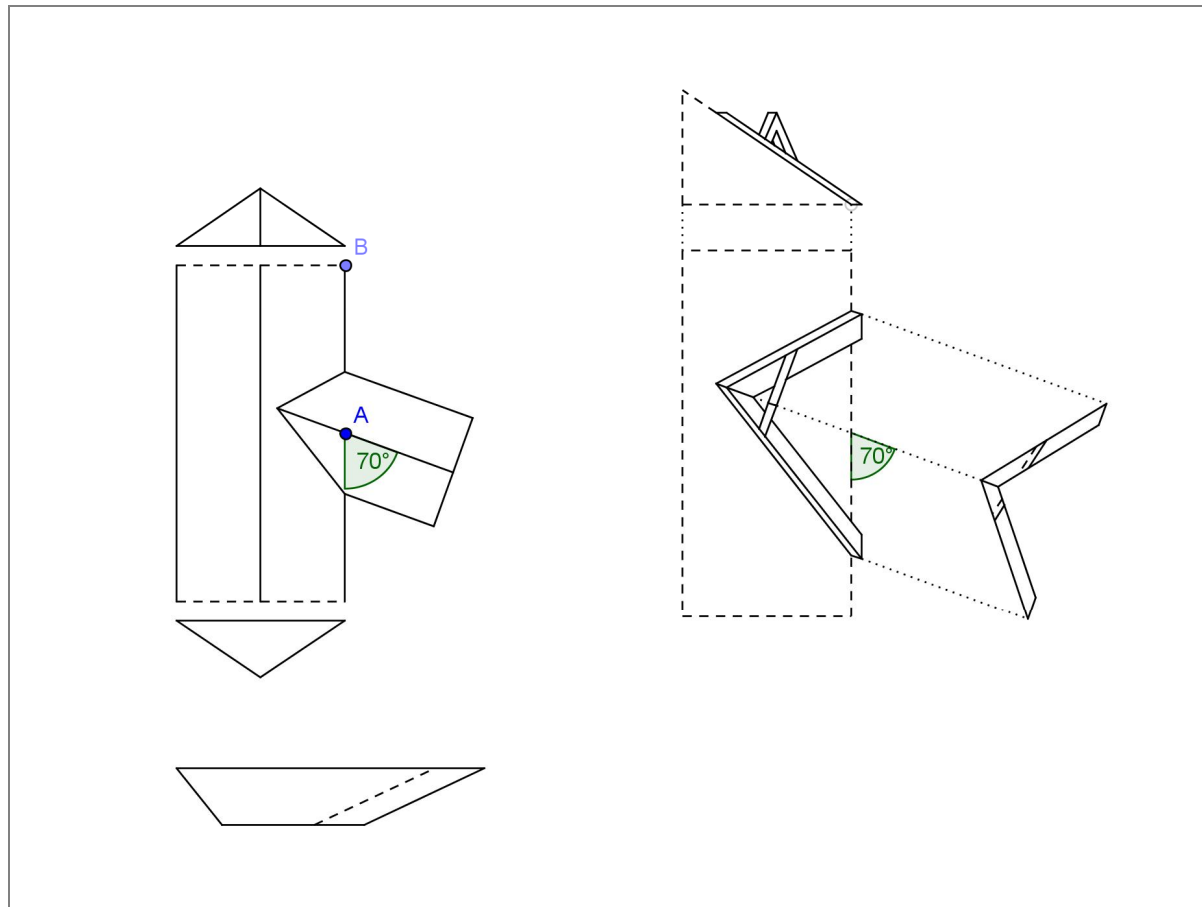
19. ábra

Első kísérleti modell:

A modell részei:

- Egy méretarányos, a fedélidom-csatlakozást szemléltető ábra (20. ábra – baloldal)
- Egy értelmező-ábra, mely a csatlakozás kérdéses elemeinek geometriáját mutatja be. Ez az ábra az elemek szélességi és magassági méreteit tekintve önmagán belül méretarányos. Az elemek hosszmeretei a baloldali szemléltető-ábra változásait követik. (20. ábra – jobboldal)

A modell hibája: Az értelmezőábra – az elemek méretváltozásai révén – bizonyos esetekben értelmezhetetlen, hibás képet mutatott.



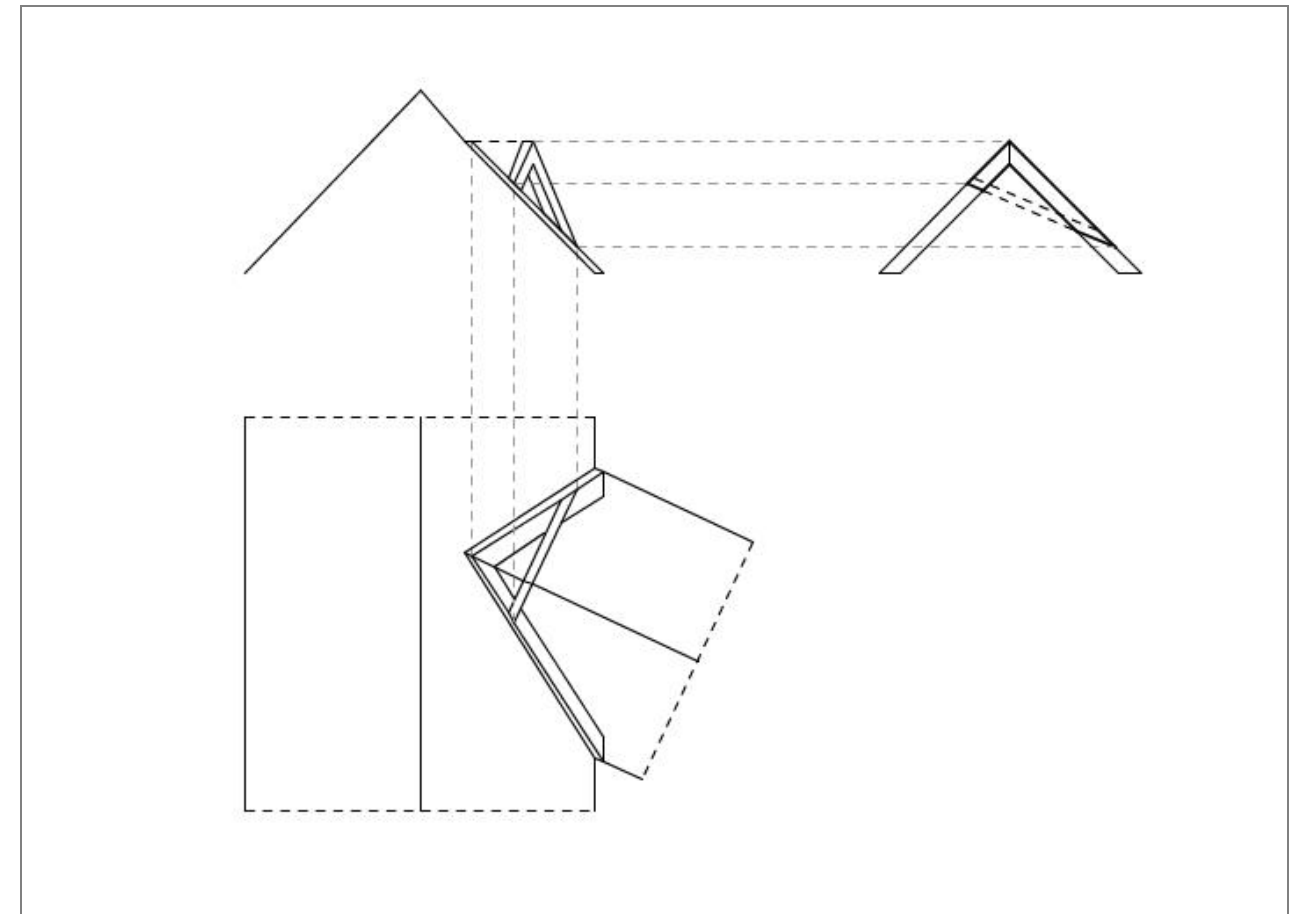
20. ábra Csonkaszaruk szerkesztése – első kísérleti modell

Második kísérleti modell:

Véghezvitt változtatások az előző modellhez képest (21. ábra):

- Elhagytam a fedélidom-csatlakozást szemléltető ábrát
- Az elemek hosszmereteit a szelvényméretektől és az aktuális szögektől tettem függővé úgy, hogy az ábra bármely szituációban helyes képet mutasson.

A modell hibája: Az értelmezőábra olykor irreálisan nagy méreteket vett fel. Továbbá a fedélidom-részlet hiánya nem teszi kellően érthetővé a szituációt.



21. ábra Csonkaszaruk szerkesztése – második kísérleti modell

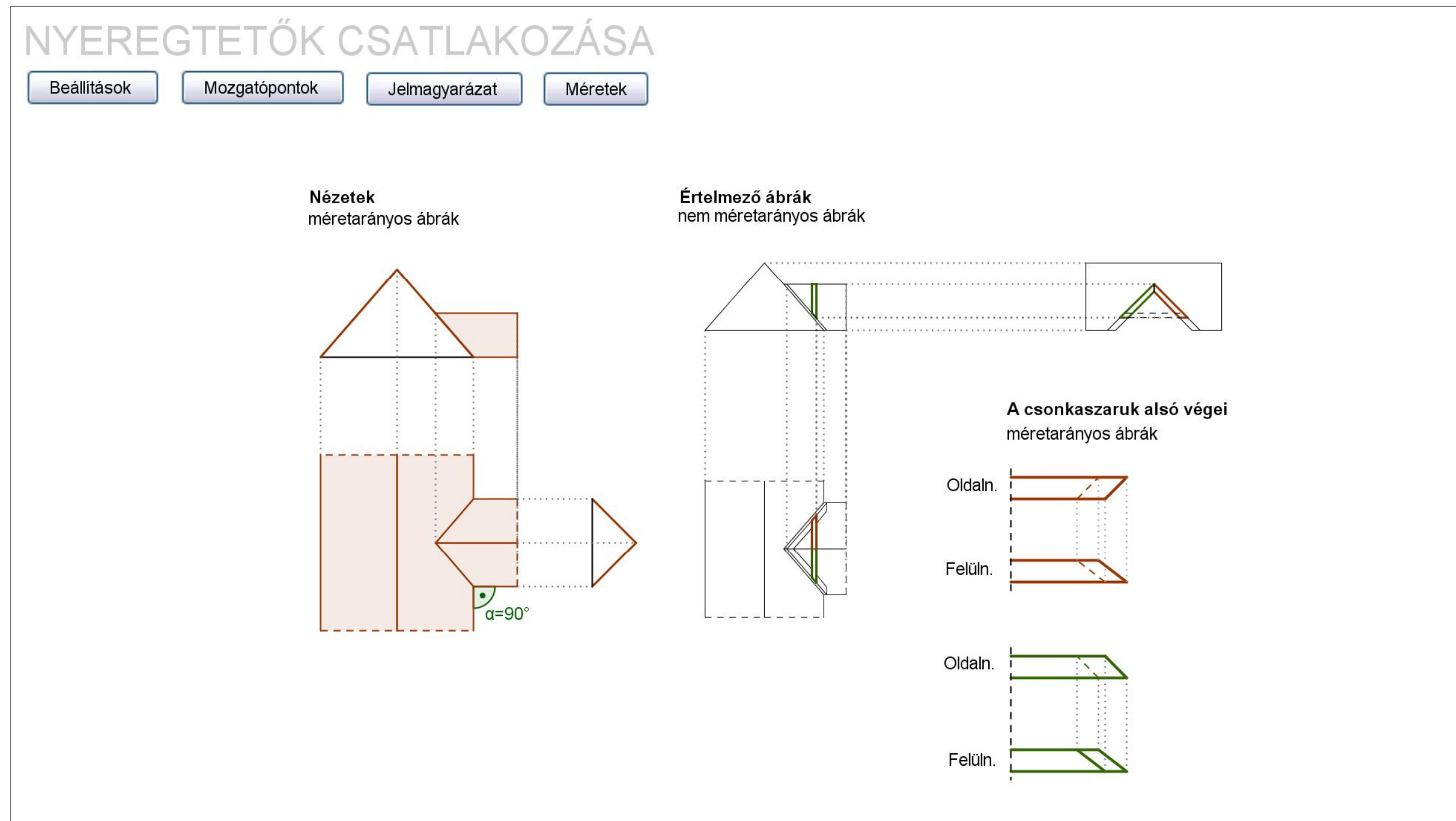
Teljes programmodell:

Véghezvitt változtatások az előző modellhez képest (22. ábra):

- Újra beépítettem a fedélidom-csatlakozást szemléltető ábrát (bal oldalt)
- Az értelmezőábra irreális méretváltozásainak korrigálására az előlnézeti rajzon lehetővé tettem a csonkaszaruk találkozási pontjának függőleges mozgását. Ez a pont más mozgatópontokkal a megfelelő gombra kattintva jelenik meg.
- Kiszerveztettem és megjelenítettem a csonkaszaruk keresett alsó végződéseit, majd színekkel jelöltem őket a könnyebb összeegyeztethetőség végett.

A Beállítások menüpont alatt megadható méretek, szögek:

- a csatlakozó fedélidomok szélessége és magassága
- a fedélidomok alaprajzon bezárt szöge (α)
- a csonkaszaruk szelvényméretei és a palló vastagsága. (A palló szélessége a csonkaszaruk méreteihez igazodik úgy, hogy azoknak teljes felfekvést biztosítson)



22. ábra Csonkaszaruk szerkesztése – harmadik (kész) változat

5. Fedélidom-szerkesztések

A fedélidom-szerkesztéssel kapcsolatos tevékenységem két fő területre terjed ki:

Egyrészt tetőközepeléseket végeztem négyszög és ötszög alaprajzon. Tetőközepelés alatt egyenlő hajlásszögekkel véghezvitt fedélidom-szerkesztést értek. Az általam készített programmodellekben az alaprajzi sarokpontok egy vízszintes síkban szabadon mozgathatók, a fedélidom felülnézete pedig folyamatosan követi az alaprajz változását, még alapvetően eltérő alaprajzi kontúrok és fedélformák esetén is. Ennek háttérében egy általam kifejlesztett algoritmus áll, melyet szerkesztéseimmel együtt mutatok be.

Másrészt elkészítettem egy komplex fedélidom-szerkesztést négyszög alaprajzon. Ezt a mozgatható sarokpontok mellett állítható dőlésszögekkel és más beépített funkciókkal láttam el. Ezeket a továbbiakban részletezem.

5.1 Tetőközepelések és az alkalmazott algoritmus kidolgozása

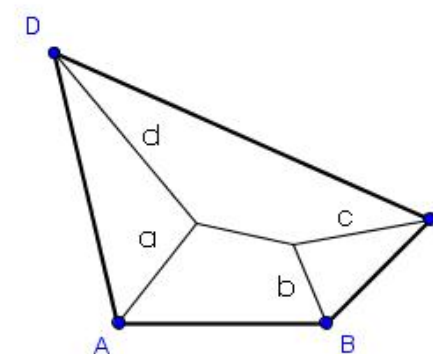
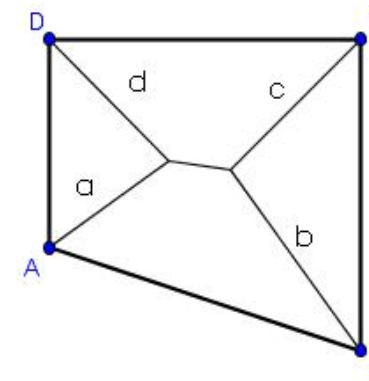
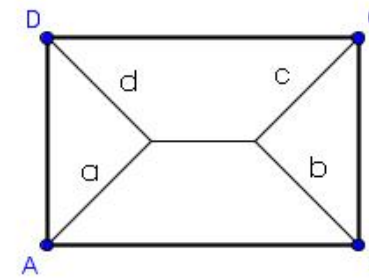
Az algoritmus lényege a következő: A program a felmerülő eseteket figyelembe véve egyszerre több módon is kiserkeszti a fedélidomot a megadott alaprajzi kontúrra. Ez után a beprogramozott feltételrendszernek megfelelően eldönti, hogy ezek közül aktuálisan melyiket jelenítse meg. Az algoritmus kidolgozására számos kísérletet végeztem. Ezek közül a fontosabbakat a következőkben mutatom be.

1. kísérleti algoritmus - négyszög alaprajzon:

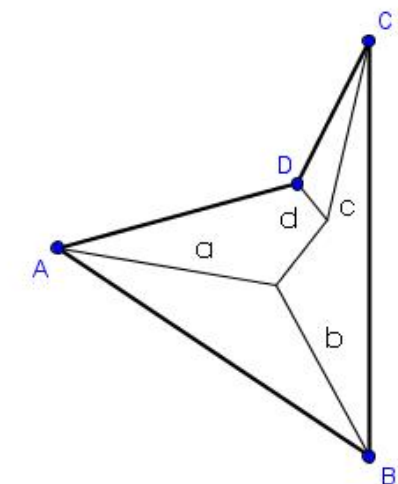
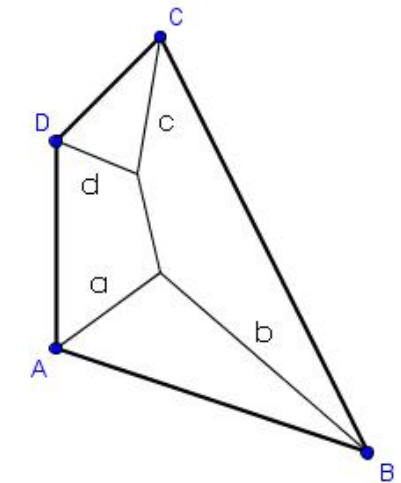
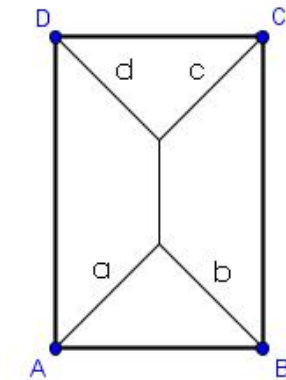
A kísérlet lépései:

1. Különböző alaprajzi kontúrokkal négyszög fedélidomokat rajzolok.
2. A megrajzolt fedélidomok sarkait az ábécé kis- és nagybetűivel jelölöm, minden fedélidomnál azonos módon és irányban haladva. (A program is hasonlóan elnevezett elemekkel dolgozik.)
3. Az így kapott fedélidomokat két fő típusba sorolom szerkesztési szisztémájuk szerint, vagyis az alapján, hogy mely élgerinc mely élgerinccel találkozik (23. ábra):
 - 1-es típus: a és d , valamint b és c él fut be egy pontba
 - 2-es típus: a és b , valamint c és d él fut be egy pontba
4. Az általam használt dinamikus geometriai programban 4 független, mozgatható sarokpontra mindkét szisztéma szerint kiserkesztem a fedélidomot.
5. A kiserkesztett, egyszerre jelen lévő két fedélidomnak (két típusnak) feltételül szabtam, hogy mikor melyik jelenjen meg. Ezt a következő feltételrendszerrel tettem:
 - 1-es típus: Ha az A -ból induló szögfelezőt a D -ből induló szögfelező metszi előbb
 - 2-es típus: Ha az A -ból induló szögfelezőt a B -ből induló szögfelező metszi előbb (mindezt az A csúcsból nézve)

1-es típus



2-es típus



23. ábra négyszög alaprajzú fedélidomok 2 típusba sorolása egyéni szempont alapján

Az 1. kísérleti algoritmus hibája:

A hiba konkáv négyszögek esetén jelentkezett (24. ábra). A megadott feltételrendszernek megfelelően a program megállapította, hogy az A -ból induló szögfelezőt (élgerincet) először a B -ből induló metszi el és ennek megfelelően azzal kötötte össze. A megadott feltételrendszer nem vette figyelembe, hogy a B -ből induló élgerinc a valóságban nem érhet el az A csúcs élgerincéig. Tehát a feltételrendszer módosításra szorult. Ezeket a módosításokat munkám során végre is hajtottam, de azok nem vezettek eredményre, így végül egy alapvetően új feltételrendszer megírásához fogtam.

2. kísérleti algoritmus – négyszög alaprajzon:

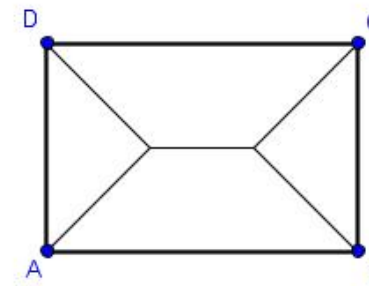
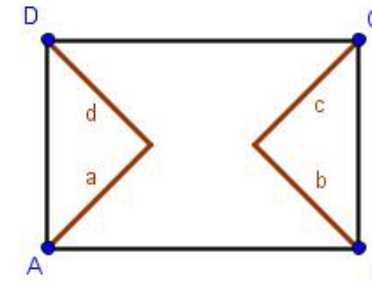
Ebben a lépcsőben egy új feltételrendszert írtam arra vonatkozóan, hogy a program mely szituációkban jelenítse meg az 1-es, és mikor a 2-es típusú kiserkesztett fedélidomot. Itt már nem a szögfelezők, hanem a rajtuk elhelyezkedő élgerincek egymással vett metszéspontjait vizsgáltam.

Az új feltételrendszer lényege (25. ábra):

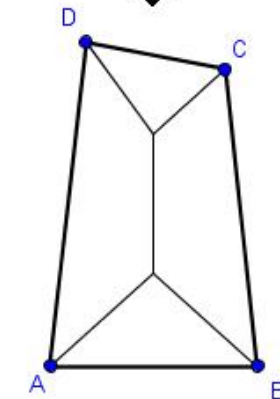
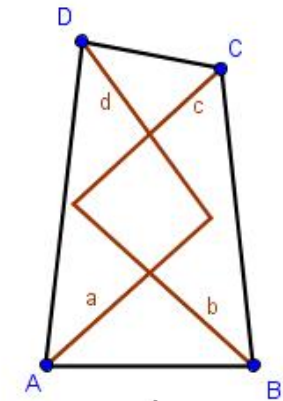
1. Vettem a két egyszerre jelen lévő, kiserkesztett fedélidomból az 1-es típust
2. Ennél a típusnál az élgerinceket párosával csoportosítottam $a-d$ és $b-c$ élgerincpárra
3. Az 1-es és 2-es típusú fedélidom megjelenésének a következő feltételt szabtam:
 - ha $a-d$ és $b-c$ élpár nem metsz egymásba, akkor az 1-es típusú fedélidom jelenjen meg
 - ha $a-d$ és $b-c$ élpár egymásba metsz, akkor a 2-es típusú fedélidom kerüljön megjelenítésre

Ez a módszer már bármely konvex és konkáv négyszög esetén helyesen működik (26. ábra). Ennek a viszonylag egyszerű szerkesztésnek a forráskódját dolgozatom végéhez csatoltam.

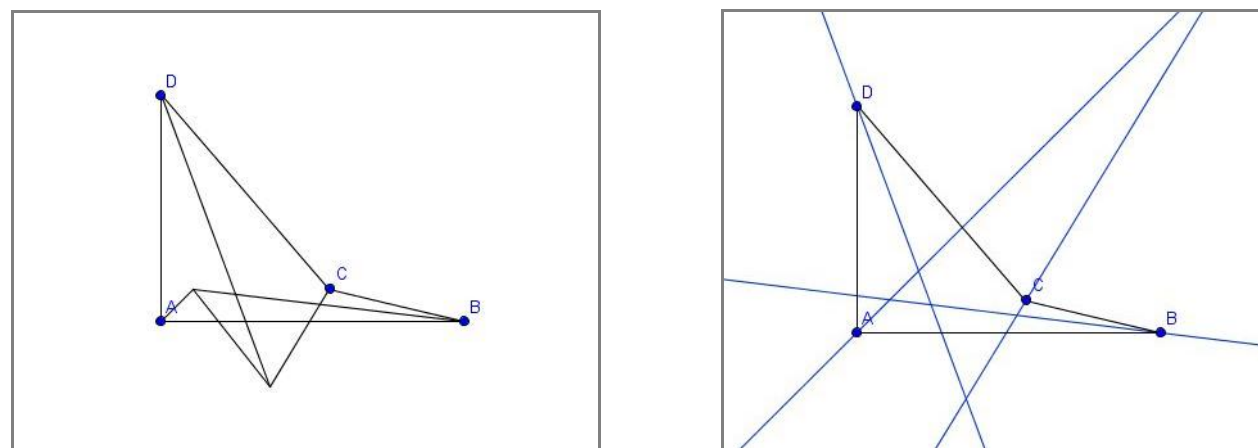
1-es típus



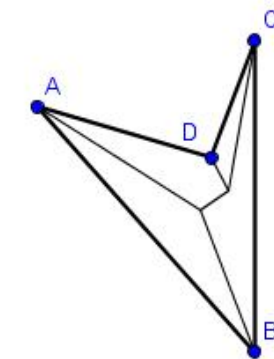
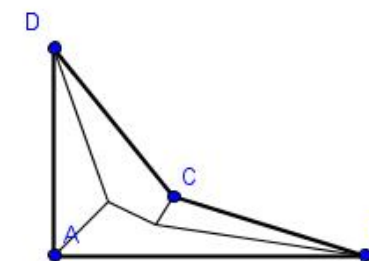
2-es típus



25. ábra tetőközepelés immár helyes algoritmussal



24. ábra az első kísérleti algoritmus hibája és a hiba háttere



26. ábra tetőközepelés konkáv négyszögeken

2. kísérleti algoritmus – ötszög alaprajzon

Több kísérlet után sikeresen alkalmaztam a második kísérleti modellt ötszög alaprajzon is.

Az algoritmus működése az előzőhöz hasonló volt:

1. Számbavettem a lehetségesen felmerülő födélidom-típusokat
2. Minden födélidom-típust kiszereztettem – itt összesen 5 típust
3. A kiszerezett öt födélidom-típusnak külön-külön feltételül szabtam, hogy mikor-melyik jelenjen meg. Példa egy ilyen feltételre:

$((I \neq I) \text{ vagy } F=G) \text{ és } ((J \neq J) \text{ vagy } F=H) \text{ és } ((K=K) \text{ vagy } H=G) \text{ és } ((L=L) \text{ vagy } F=G \text{ vagy } F=H)$
és $((M=M \text{ vagy } F=G \text{ vagy } G=H)$

Mint látható, egy ilyen feltétel és igen összetett. Ugyanezen szisztéma alkalmazása 6- és magasabb oldalszámú alaprajzi kontúrok esetén hatványozottan megnöveli a szerkesztésbe fektetendő időt és energiát, ezért úgy döntöttem, hogy egy új, egyszerűbb feltételrendszert keresek.

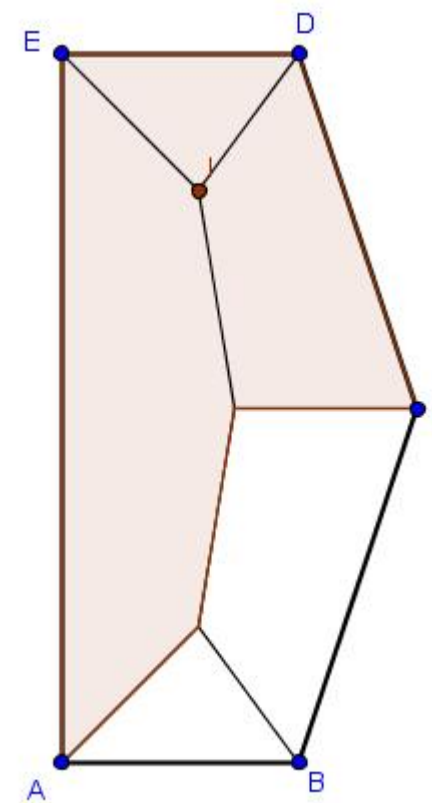
Végleges algoritmus – ötszög alaprajzon

Több kísérletet is végeztem, hogy találjak egy új, röviden leírható feltételrendszert szabálytalan sokszög alaprajzon végzett tetőközepeléseimhez. Végül kísérletezésem eredményre vezettek a program újonnan megjelent 4.0-s verziójával. Ebben számos új parancs volt elérhető, amely a korábbi verziókban nem. Ilyen például az *IsInregion*. Ennek a parancsnak a segítségével meg lehetett vizsgálni, hogy egy adott pont egy megadott sokszögön belül helyezkedik-e el. Egy megadott változó ettől függően vesz fel *igaz* vagy *hamis* értéket. Ezt felhasználva egy új, viszonylag egyszerű feltételrendszert hoztam létre.

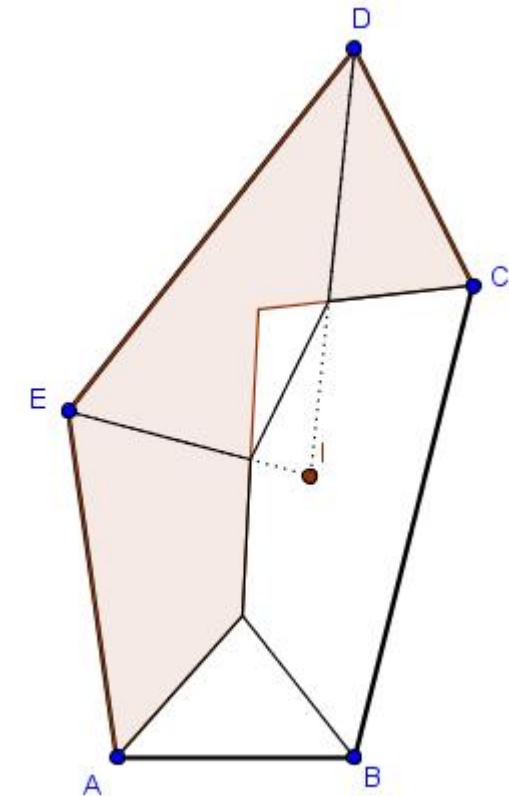
Az új feltételrendszer alapja:

Mint eddig, a program itt is azt vizsgálja, hogy egy adott, a háttérben elvégzett szerkesztés eredményének megjelenítésével helyes ábra jön-e létre. Itt ezt úgy teszi, hogy megnézi, hogy az élgerincek metszéspontjai egy megengedett tartományon belül maradnak-e (27. a. ábra). Ha valamely csomópont kiesik a hozzá rendelt tartományból, akkor egy másik, a háttérben elvégzett szerkesztés (tetőközepelés) eredménye kerül megjelenítésre (27. b. ábra – fekete vonallal jelölve az aktuálisan megjelenített fedélidom).

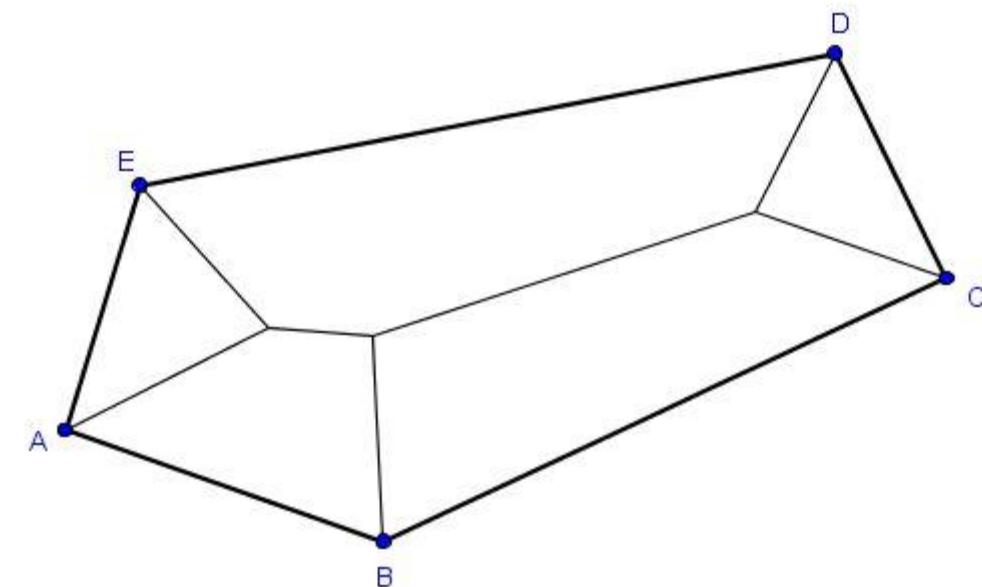
Ez a z algoritmus a vele alkalmazott feltételrendszerrel egyszerűsége mellett valamennyi esetben helyes szerkesztési eredményt ad. Továbbá alkalmasnak mutatkozik a jövőben magasabb oldalszámú fedélidomok szerkesztésére.



27. a. ábra egy találkozási pont a hozzá rendelt tartományon belül



27. b. ábra egy találkozási pont a hozzá rendelt tartományon kívül



28. ábra egy helyes tetőközepelés ötszög alaprajzon

5.2 Komplex fedélidom-szerkesztés

Lényeges különbség előzően bemutatott tetőközepeléseimhez képest, hogy ez a szerkesztés állítható tető-hajlásszögekkel és beépített nézettel rendelkezik. Meg kell jegyeznem, hogy a nézeten a láthatóság feltüntetése még nem megoldott. A szerkesztést egyelőre négyszög alaprajzi kontúron valósítottam meg.

További beépített funkciók – az őket megjelenítő menüpontok szerint csoportosítva:

Alaprajz:

- Számszerűen és csúszkákkal is állítható alaprajzi szögek és hosszak (29. ábra)
- Forgatható alaprajz az A sarokpont körül α szög állításával.

Nézet:

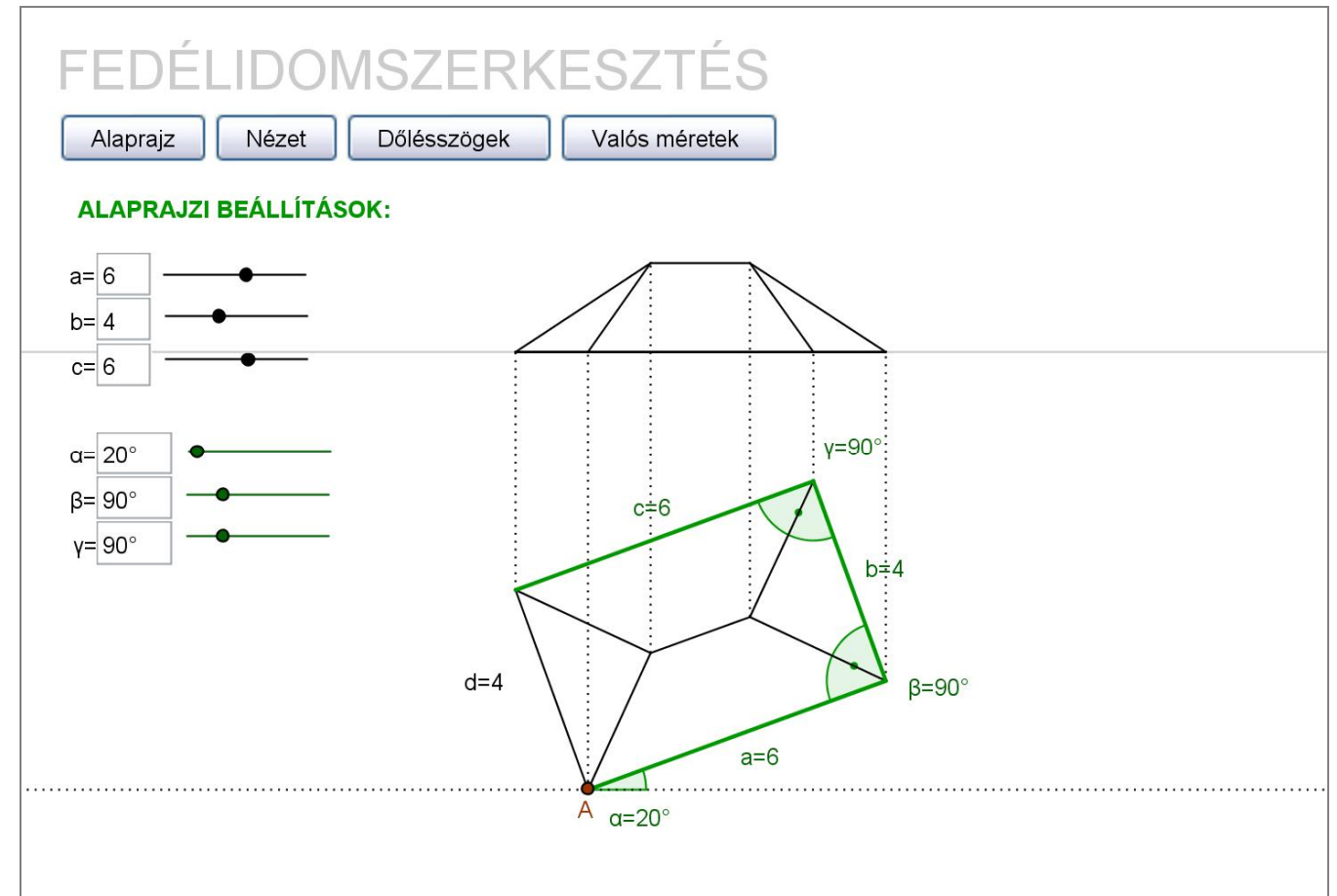
- A nézet forgatása az alaprajz súlypontja körül
- Egy fogópont segítségével állítható a nézet alaprajztól mért távolsága

Dőlésszögek:

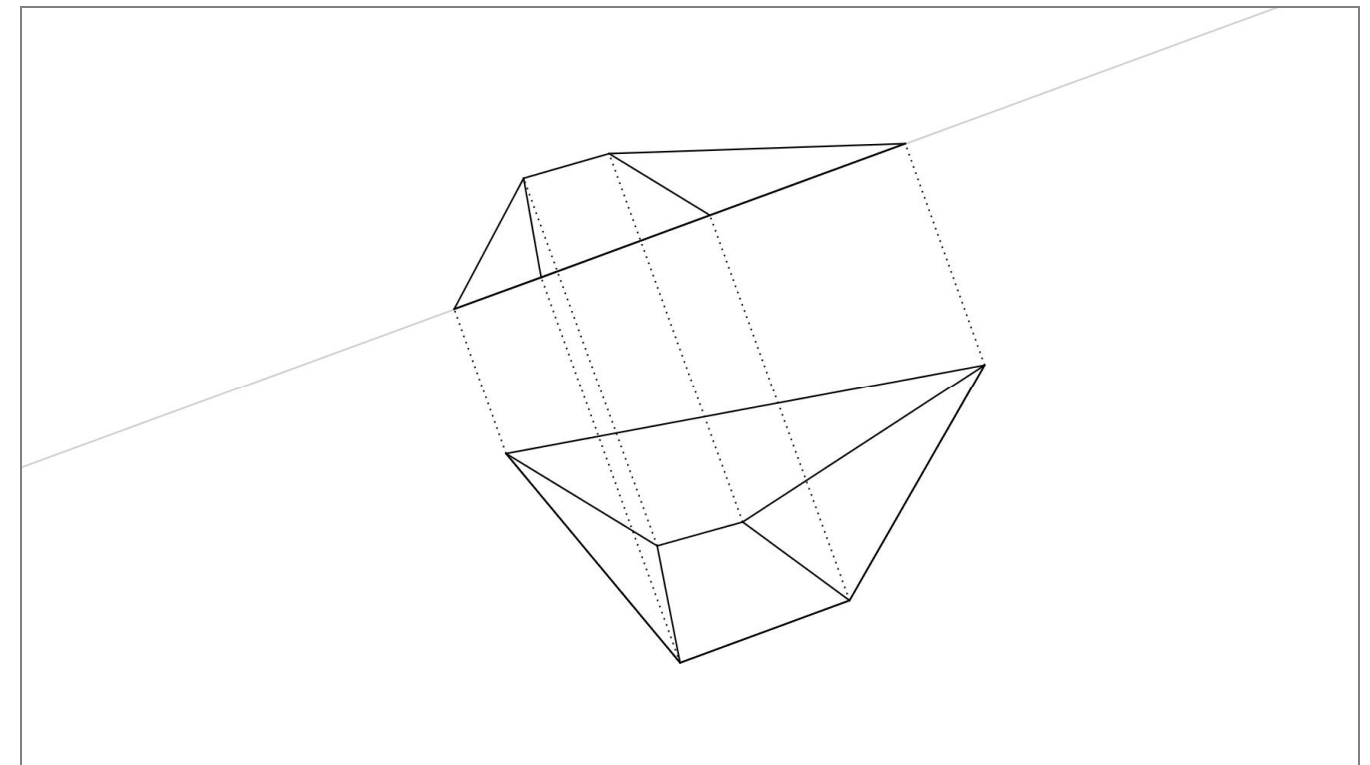
- Állítható tető-dőlésszögek

Valós méretek:

- Az élek, vápák és a gerinc hosszainak megjelenítése
- Az oldalak valós területeinek megjelenítése



29. ábra komplex fedélidom-szerkesztés – beállítások



30. ábra komplex fedélidom-szerkesztés

6. Összegzés

6.1 Elért eredmények

Létrehoztam egy rendszert, mely adott, a dolgozatban tárgyalt szituációkban képes segítséget nyújtani az ácsok, építészek számára tetőszerkezeti és egyéb ácsszerkezeti részletek kidolgozásában.

Ezen belül:

- Programmodellek formájában konkrét szituációkra adaptálható ácsszerkezeti részleteket dolgoztam ki
- Ezeket a modelleket egy könnyen átlátható kezelőfelülettel, adatbeviteli panellel láttam el
- Végül bárki számára elérhetővé tettem őket egy honlapon keresztül

Bemutattam annak a lehetőségét, hogy a dinamikus geometriai programok hogyan használhatók szemléltetőeszközként a fedélidom-szerkesztésekkel kapcsolatos síkbeli és térbeli feladatokban, akár az oktatás területén is.

Ennek részeként:

- Egy dinamikus geometriai programban adott oldalszámú, szabadon alakítható alaprajzokon fedélidom-szerkesztéseket végeztem el
- Ezeket a szerkesztéseket az előzően említett szerkezeti részletekkel együtt elérhetővé tettem egy honlap keretében

6.2 Rendszerkövetelmények, felhasználhatóság

A honlap jelenleg a következő címeiről érhető el:

- users.hszk.bme.hu/~kp720
- http://www.epab.bme.hu/TDK_munkak/kondorp/2011

Az elvégzett szerkesztések letöltéséhez szükséges:

- Internet-hozzáférés
- tetszőleges típusú böngésző program

A szerkesztések rendszerkövetelménye:

- Windows, MAC OS X, vagy GNU grafikus felülettel ellátott Linux operációs rendszer
- megfelelő verziójú Java szoftver (bárki számára ingyenesen letölthető)

A szerkesztési eredmények nyomtatása:

A program lehetővé teszi a kiszerkesztett ábrák nyomtatását, de méretarányos nyomtatási lehetőséget még nem biztosít. Mindazonáltal a szerkesztési eredmények a kiírt méretek után pár perc alatt papírra vethetők, vagy az elemvégződésekről adott esetben méretarányos sablon készíthető.

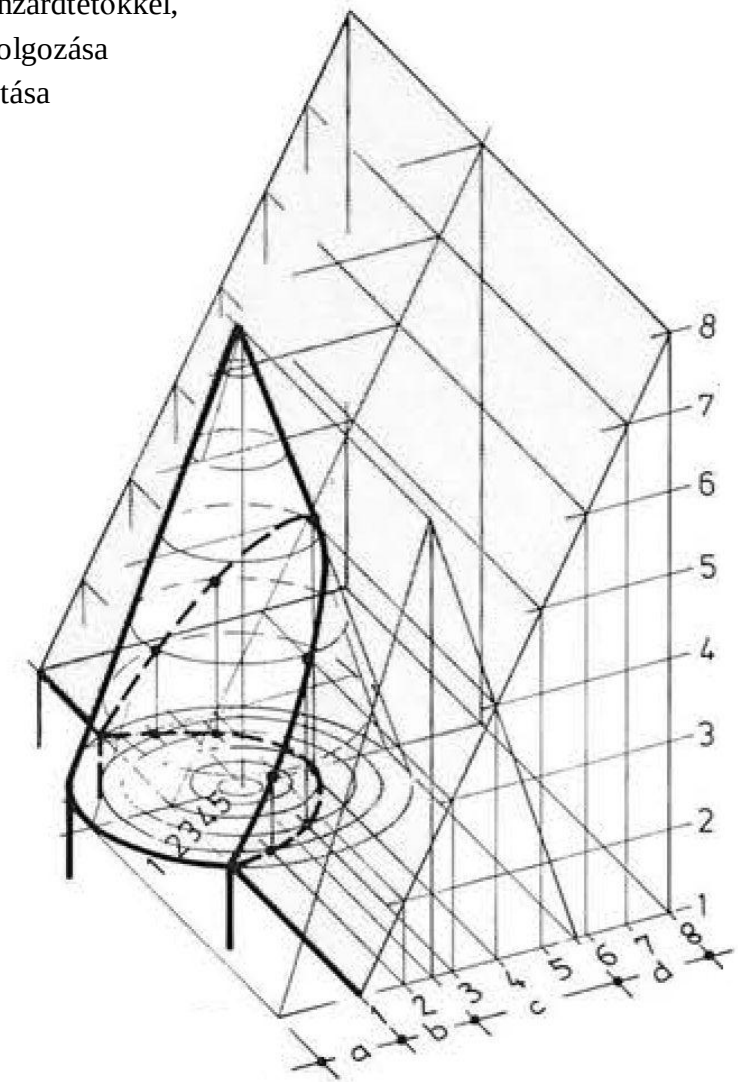
6.3 További terveim

Időmhöz, lehetőségeimhez és az esetlegesen felmerülő igényekhez mérten a jövőben szeretném a következő fejlesztéseket véghezvinni:

- láthatóság feltüntetése a fedélidomok nézetein
- nagyobb - 6, 7, stb. - oldalszámú szabályos és szabálytalan fedélidomok kiszerkesztése
- a részletrajzok esetében az elemvégzések mellett szabási sablonok kiszerkesztése
- további, a programban elérhető funkciók tanulmányozása, a szerkesztésekhez való esetleges felhasználása, pl: CAS, táblázatkezelő, scriptek
- áttérés a 3D-s szerkesztést is lehetővé Geogebra 5.0-ra

Tervezett további szerkesztéseim:

- toronytetők fedélidomainak szerkesztése
- fedélidom-szerkesztés manzárdtetőkkel,
- új szerkezeti részletek kidolgozása
- kúptető és nyeregtető áthatása



7. Irodalomjegyzék

- | | |
|---|---|
| [1] Dachkonstruktionen in Holz | Willibald Mannes, Deutsche Verlags - Anstalt DVA, 2001 |
| [2] Fejezetek az épületszerkezetek köréből /
1. füzet - Födélszerkezetek és födél szerelvények / 1. rész | Dr. Sándy Gyula |
| [3] Épületszerkezettan | Dr. Kotsis Endre, Egyetemi Nyomda, Budapest, 1945. |
| [4] Ácsszerkezetek | Tóbiás László és Tóbiás Lóránt, Budapest, Műszaki Könyvkiadó, 1950. |
| [5] Geogebra – fórum | www.geogebra.org/forum |
| [6] Debreceni műszaki közlemények 2009/1-2:
A dinamikus geometriai rendszerek és az ábrázoló geometria | Nagyné Dr. Kondor Rita, Debreceni Egyetem, Műszaki Kar - www.mfk.unideb.hu |

8. Egy szerkesztés forráskódja

- 5.1 fejezet - 2. kísérleti algoritmus négyszög alaprajzon alkalmazva

```
<?xml version="1.0" encoding="utf-8"?>
<geogebra format="4.0" id="ab7a6712-de45-40d5-9097-f552bdbfa854"
xsi:noNamespaceSchemaLocation="http://www.geogebra.org/ggb.xsd"
xmlns="" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
<gui>
  <window width="1024" height="738" />
  <perspectives>
<perspective id="tmp">
  <panes>
    <pane location="" divider="0.2470703125" orientation="1" />
  </panes>
  <views>
    <view id="4" toolbar="0 || 2020 , 2021 , 2022 , 66 || 2001 ,
2003 , 2002 , 2004 , 2005 || 2040 , 2041 , 2042 , 2044 , 2043" visible="false"
inframe="false" stylebar="false" location="1,1" size="200"
window="0,0,400,400" />
    <view id="1" visible="true" inframe="false" stylebar="false"
location="1" size="1016" window="0,0,400,400" />
    <view id="2" visible="false" inframe="false" stylebar="false"
location="3" size="253" window="0,0,200,400" />
    <view id="16" visible="false" inframe="true" stylebar="false"
location="1" size="150" window="50,50,500,500" />
    <view id="32" visible="false" inframe="true" stylebar="true"
location="1" size="150" window="50,50,500,500" />
    <view id="64" visible="false" inframe="true" stylebar="false"
location="1" size="150" window="50,50,500,500" />
  </views>
  <toolbar show="true" items="0 39 59 || 1 501 67 , 5 19 , 72 | 2 15 45 ,
18 65 , 7 37 | 4 3 8 9 , 13 44 , 58 , 47 || 16 51 64 , 70 | 10 34 53 11 , 24 20 22 , 21
23 | 55 56 57 , 12 || 36 46 , 38 49 50 , 71 | 30 29 54 32 31 33 | 17 26 62 , 14 66 68
| 25 52 60 61 || 40 41 42 , 27 28 35 , 6" />
  <input show="true" cmd="true" top="false" />
</perspective>
  </perspectives>
  <labelingStyle val="3"/>
  <font size="12"/>
  <graphicsSettings javaLatexFonts="false"/>
  <consProtColumns col0="true" col1="true" col2="false" col3="true"
col4="false" col5="true" col6="true" col7="false"/>
  <consProtocol useColors="true" addIcons="false"
showOnlyBreakpoints="false"/>
</gui>
<euclidianView>
  <size width="1016" height="575"/>
  <coordSystem xZero="287.61614286111876"
yZero="319.4474955922244" scale="53.0935057929429"
yscale="53.093505792942885"/>
  <evSettings axes="false" grid="false" gridIsBold="false"
pointCapturing="3" rightAngleStyle="1" checkboxSize="13" gridType="0"/>
  <bgColor r="255" g="255" b="255"/>
  <axesColor r="0" g="0" b="0"/>
  <gridColor r="192" g="192" b="192"/>
  <lineStyle axes="1" grid="10"/>
  <axis id="0" show="false" label="" unitLabel="" tickStyle="1"
showNumbers="true"/>
```

```
  <axis id="1" show="false" label="" unitLabel="" tickStyle="1"
showNumbers="true"/>
</euclidianView>
<kernel>
  <continuous val="false"/>
  <decimals val="2"/>
  <angleUnit val="degree"/>
  <algebraStyle val="0"/>
  <coordStyle val="0"/>
  <angleFromInvTrig val="false"/>
</kernel>
<scripting blocked="false" disabled="false"/>
<construction title="" author="" date="">
<element type="point" label="A">
  <show object="true" label="true"/>
  <objColor r="0" g="0" b="255" alpha="0.0"/>
  <layer val="0"/>
  <labelMode val="0"/>
  <animation step="0.1" speed="1" type="0" playing="false"/>
  <coords x="1.0" y="1.0" z="1.0"/>
  <pointSize val="3"/>
  <pointStyle val="0"/>
</element>
<element type="point" label="B">
  <show object="true" label="true"/>
  <objColor r="0" g="0" b="255" alpha="0.0"/>
  <layer val="0"/>
  <labelMode val="0"/>
  <animation step="0.1" speed="1" type="0" playing="false"/>
  <coords x="3.0" y="1.0" z="1.0"/>
  <pointSize val="3"/>
  <pointStyle val="0"/>
</element>
<element type="point" label="C">
  <show object="true" label="true"/>
  <objColor r="0" g="0" b="255" alpha="0.0"/>
  <layer val="0"/>
  <labelMode val="0"/>
  <animation step="0.1" speed="1" type="0" playing="false"/>
  <coords x="4.0" y="2.0" z="1.0"/>
  <pointSize val="3"/>
  <pointStyle val="0"/>
</element>
<element type="point" label="D">
  <show object="true" label="true"/>
  <objColor r="0" g="0" b="255" alpha="0.0"/>
  <layer val="0"/>
  <labelOffset x="-19" y="-3"/>
  <labelMode val="0"/>
  <animation step="0.1" speed="1" type="0" playing="false"/>
  <coords x="0.38392373670662044" y="3.6058552309361978"
z="1.0"/>
  <pointSize val="3"/>
  <pointStyle val="0"/>
</element>
```

```
<command name="Segment">
  <input a0="A" a1="B"/>
  <output a0="a"/>
</command>
<element type="segment" label="a">
  <show object="true" label="false"/>
  <objColor r="0" g="0" b="0" alpha="0.0"/>
  <layer val="0"/>
  <labelMode val="0"/>
  <coords x="0.0" y="2.0" z="-2.0"/>
  <lineStyle thickness="4" type="0" typeHidden="1"/>
  <eqnStyle style="implicit"/>
  <outlyingIntersections val="false"/>
  <keepTypeOnTransform val="true"/>
</element>
<command name="Segment">
  <input a0="B" a1="C"/>
  <output a0="b"/>
</command>
<element type="segment" label="b">
  <show object="true" label="false"/>
  <objColor r="0" g="0" b="0" alpha="0.0"/>
  <layer val="0"/>
  <labelMode val="0"/>
  <coords x="-1.0" y="1.0" z="2.0"/>
  <lineStyle thickness="4" type="0" typeHidden="1"/>
  <eqnStyle style="implicit"/>
  <outlyingIntersections val="false"/>
  <keepTypeOnTransform val="true"/>
</element>
<command name="Segment">
  <input a0="C" a1="D"/>
  <output a0="c"/>
</command>
<element type="segment" label="c">
  <show object="true" label="false"/>
  <objColor r="0" g="0" b="0" alpha="0.0"/>
  <layer val="0"/>
  <labelMode val="0"/>
  <coords x="-1.6058552309361978" y="-3.6160762632933796"
z="13.65557345033155"/>
  <lineStyle thickness="4" type="0" typeHidden="1"/>
  <eqnStyle style="implicit"/>
  <outlyingIntersections val="false"/>
  <keepTypeOnTransform val="true"/>
</element>
<command name="Segment">
  <input a0="D" a1="A"/>
  <output a0="d"/>
</command>
<element type="segment" label="d">
  <show object="true" label="false"/>
  <objColor r="0" g="0" b="0" alpha="0.0"/>
  <layer val="0"/>
  <labelMode val="0"/>
```

```

        <coords x="2.6058552309361978" y="0.6160762632933796" z="-
3.2219314942295774"/>
        <lineStyle thickness="4" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
        <outlyingIntersections val="false"/>
        <keepTypeOnTransform val="true"/>
    </element>
    <command name="AngularBisector">
        <input a0="A" a1="B" a2="C"/>
        <output a0="e"/>
    </command>
    <element type="line" label="e">
        <show object="false" label="true"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="-0.9238795325112867" y="-0.3826834323650897"
z="3.1543220298989496"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
    </element>
    <command name="AngularBisector">
        <input a0="B" a1="C" a2="D"/>
        <output a0="f"/>
    </command>
    <element type="line" label="f">
        <show object="false" label="true"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="0.18270381676073172" y="-0.9831679995509724"
z="1.235520732059018"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
    </element>
    <command name="AngularBisector">
        <input a0="C" a1="D" a2="A"/>
        <output a0="g"/>
    </command>
    <element type="line" label="g">
        <show object="false" label="true"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="0.7696436158598564" y="0.6384737305215977" z="-
2.597728293949962"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
    </element>
    <command name="AngularBisector">
        <input a0="D" a1="A" a2="B"/>
        <output a0="h"/>
    </command>
    <element type="line" label="h">
        <show object="false" label="true"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="-0.7842440283772389" y="0.6204524993540124"
z="0.16379152902322658"/>

```

```

        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
    </element>
    <command name="Intersect">
        <input a0="e" a1="h"/>
        <output a0="E"/>
    </command>
    <element type="point" label="E">
        <show object="false" label="true"/>
        <objColor r="64" g="64" b="64" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="2.0197872917371593" y="2.3224345742637342"
z="0.8733405616398711"/>
        <pointSize val="3"/>
        <pointStyle val="0"/>
    </element>
    <command name="Intersect">
        <input a0="g" a1="h"/>
        <output a0="F"/>
    </command>
    <element type="point" label="F">
        <show object="false" label="true"/>
        <objColor r="64" g="64" b="64" alpha="0.0"/>
        <layer val="0"/>
        <labelOffset x="1" y="0"/>
        <labelMode val="0"/>
        <coords x="1.7163436011871847" y="1.9111917972322"
z="0.9782465155094088"/>
        <pointSize val="3"/>
        <pointStyle val="0"/>
    </element>
    <command name="Intersect">
        <input a0="f" a1="g"/>
        <output a0="G"/>
    </command>
    <element type="point" label="G">
        <show object="false" label="true"/>
        <objColor r="64" g="64" b="64" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="1.765155799205248" y="1.4255255179037216"
z="0.873340561639871"/>
        <pointSize val="3"/>
        <pointStyle val="0"/>
    </element>
    <command name="Intersect">
        <input a0="e" a1="f"/>
        <output a0="H"/>
    </command>
    <element type="point" label="H">
        <show object="false" label="true"/>
        <objColor r="64" g="64" b="64" alpha="0.0"/>
        <layer val="0"/>
        <labelOffset x="9" y="8"/>
        <labelMode val="0"/>
        <coords x="2.62841516557274" y="1.7177789904976852"
z="0.9782465155094087"/>
        <pointSize val="3"/>
        <pointStyle val="0"/>

```

```

    </element>
    <command name="Segment">
        <input a0="E" a1="G"/>
        <output a0="i"/>
    </command>
    <element type="segment" label="i">
        <show object="true" label="false"/>
        <condition showObject="(J ≠ F) ∧ (I ≠ H)"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="1.026986602655769" y="-0.29156036455445244" z="-
1.5997935731724038"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
        <outlyingIntersections val="false"/>
        <keepTypeOnTransform val="true"/>
    </element>
    <command name="Segment">
        <input a0="F" a1="H"/>
        <output a0="j"/>
    </command>
    <element type="segment" label="j">
        <show object="true" label="false"/>
        <condition showObject="J ≧ F ∨ I ≧ H"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="0.19771377016742786" y="0.9323535018273041" z="-
2.1684218604161405"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
        <outlyingIntersections val="false"/>
        <keepTypeOnTransform val="true"/>
    </element>
    <command name="Segment">
        <input a0="A" a1="E"/>
        <output a0="k"/>
    </command>
    <element type="segment" label="k">
        <show object="true" label="false"/>
        <condition showObject="(J ≠ F) ∧ (I ≠ H)"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="-1.6592542202584752" y="1.312714398544144"
z="0.3465398217143312"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
        <outlyingIntersections val="false"/>
        <keepTypeOnTransform val="true"/>
    </element>
    <command name="Segment">
        <input a0="B" a1="E"/>
        <output a0="l"/>
    </command>
    <element type="segment" label="l">
        <show object="true" label="false"/>
        <condition showObject="(J ≠ F) ∧ (I ≠ H)"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>

```

```

        <layer val="0"/>
        <labelOffset x="-22" y="22"/>
        <labelMode val="0"/>
        <coords x="-1.6592542202584752" y="-0.6872856014558559"
z="5.665048262231282"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
        <outlyingIntersections val="false"/>
        <keepTypeOnTransform val="true"/>
    </element>
    <command name="Segment">
        <input a0="C" a1="G"/>
        <output a0="m"/>
    </command>
    <element type="segment" label="m">
        <show object="true" label="false"/>
        <condition showObject="(J ≠ F) ∧ (I ≠ H)"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="0.3677323823972938" y="-1.9788459660103084"
z="2.4867624024314416"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
        <outlyingIntersections val="false"/>
        <keepTypeOnTransform val="true"/>
    </element>

        <input a0="F" a1="D"/>
        <output a0="r"/>
    </command>
    <element type="segment" label="r">
        <show object="false" label="false"/>

        <condition showObject="J ≐ F ∨ I ≐ H"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelOffset x="-15" y="31"/>
        <labelMode val="0"/>
        <coords x="-1.652163838294771" y="-1.3705865773867774"
z="5.5764416939707315"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
        <outlyingIntersections val="false"/>
        <keepTypeOnTransform val="true"/>
    </element>
    <command name="Segment">
        <input a0="H" a1="C"/>
        <output a0="s"/>
    </command>
    <element type="segment" label="s">

```

```

    <command name="Segment">
        <input a0="D" a1="G"/>
        <output a0="n"/>
    </command>
    <element type="segment" label="n">
        <show object="true" label="false"/>
        <condition showObject="(J ≠ F) ∧ (I ≠ H)"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelOffset x="25" y="-6"/>
        <labelMode val="0"/>
        <coords x="1.9735876133334915" y="1.6372302972830712" z="-
6.6613225629342825"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
        <outlyingIntersections val="false"/>
        <keepTypeOnTransform val="true"/>
    </element>
    <command name="Segment">
        <input a0="B" a1="H"/>
        <output a0="p"/>
    </command>
    <element type="segment" label="p">
        <show object="false" label="false"/>

        <condition showObject="J ≐ F ∨ I ≐ H"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <show object="false" label="false"/>

        <condition showObject="J ≐ F ∨ I ≐ H"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="-0.2440223775260011" y="1.313136184079298" z="-
1.6501828580545919"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
        <outlyingIntersections val="false"/>
        <keepTypeOnTransform val="true"/>
    </element>
    <command name="Intersect">
        <input a0="l" a1="m"/>
        <output a0="I"/>
    </command>
    <element type="point" label="I">
        <show object="false" label="true"/>
        <objColor r="204" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelOffset x="-5" y="-2"/>
        <labelMode val="0"/>

```

```

        <labelMode val="0"/>
        <coords x="-0.7559776224739989" y="-0.3131361840792981"
z="2.5810690515012946"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
        <outlyingIntersections val="false"/>
        <keepTypeOnTransform val="true"/>
    </element>
    <command name="Segment">
        <input a0="A" a1="F"/>
        <output a0="q"/>
    </command>
    <element type="segment" label="q">
        <show object="false" label="false"/>

        <condition showObject="J ≐ F ∨ I ≐ H"/>
        <objColor r="0" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelMode val="0"/>
        <coords x="-0.9536913926414268" y="0.7545103140933977"
z="0.19918107854802902"/>
        <lineStyle thickness="2" type="0" typeHidden="1"/>
        <eqnStyle style="implicit"/>
        <outlyingIntersections val="false"/>
        <keepTypeOnTransform val="true"/>
    </element>
    <command name="Segment">

        <coords x="9.501141907537177" y="6.2093927048804325"
z="3.5361456919547822"/>
        <pointSize val="3"/>
        <pointStyle val="0"/>
    </element>
    <command name="Intersect">
        <input a0="k" a1="n"/>
        <output a0="J"/>
    </command>
    <element type="point" label="J">
        <show object="false" label="true"/>
        <objColor r="204" g="0" b="0" alpha="0.0"/>
        <layer val="0"/>
        <labelOffset x="-13" y="-11"/>
        <labelMode val="0"/>
        <coords x="9.31177953703659" y="10.36890087538951"
z="5.307338157113221"/>
        <pointSize val="3"/>
        <pointStyle val="0"/>
    </element>
</construction>
</geogebra>

```